

TÉLÉ-UNIVERSITÉ - UNIVERSITÉ DU QUÉBEC À MONTRÉAL
UNIVERSITÉ PIERRE ET MARIE CURIE – PARIS VI

MODÉLISATION DE L'APPRENANT AVEC UNE APPROCHE PAR COMPÉTENCES
DANS LE CADRE D'ENVIRONNEMENTS D'APPRENTISSAGE EN LIGNE

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE DU
DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
LUCIE MOULET

JUILLET 2011



r-libre.telug.ca/824/Moulet.pdf

REMERCIEMENTS

Je tiens à remercier et à exprimer ma profonde reconnaissance à toutes les personnes qui m'ont aidée, orientée et soutenue durant ces années de doctorat.

Je remercie particulièrement mes trois directeurs de thèse, Richard Hotte, Jean-Marc Labat et Olga Marino, pour leur confiance, leurs encouragements et pour le temps qu'ils ont consacré à me guider tout au long de ce cheminement. Je leur suis particulièrement reconnaissante pour leurs critiques et conseils toujours opportuns qui ont permis d'améliorer d'une part la recherche mais aussi la rédaction de la thèse.

Je remercie tous les membres du jury, Monique Grandbastien, Josianne Basque et Roger Nkambou, pour la lecture de cette thèse et particulièrement pour avoir accepté de porter leur jugement d'expert sur mes travaux de recherche et me fournir ainsi des suggestions pertinentes. Je remercie aussi le directeur du programme de doctorat en informatique cognitive pour son suivi académique, ainsi que l'université Pierre et Marie Curie (Paris 6) d'avoir accepté de m'accueillir en cotutelle de thèse pour la dernière année.

Je tiens aussi à remercier les membres du centre de recherche LICEF, particulièrement mes collègues doctorants (ou non) pour leur soutien et leurs conseils aussi bien intellectuels que personnels et pour les fous rire qui effacent les larmes, Julien, Délia, Alexis et Valéry. Je remercie aussi les chercheurs avec qui j'ai travaillé au cours de cette thèse, Michel Léonard, Karin Lundgren-Cayrol et Diane Ruelland. Merci à Mihai Tabaras et à Karim Ait-Seddik d'avoir toujours trouvé une réponse à mes problèmes d'ordre technique.

Merci aussi aux âmes généreuses qui ont accepté de prêter leur nom aux acteurs fictifs de mes simulations.

Je reconnais aussi la chance que j'ai eue d'obtenir une bourse de doctorat du Conseil de recherche en sciences humaines du Canada (CRSH). Je remercie Gilbert Paquette et Olga Marino de m'avoir accordé une bourse dans le cadre du projet de

recherche LORNET. Je remercie également les professeurs qui m'ont accordé leur confiance en m'engageant dans leurs projets de recherche ou dans leurs projets pédagogiques et qui m'ont permis de financer cette thèse, Josianne Basque, Béatrice Pudelko, France Henri, Daniel Lemire, Richard Hotte et Violaine Page-Lamarche. Sans ces financements la réussite de ce doctorat aurait été très difficile.

Finalement, je remercie mes parents qui ont toujours cru en moi (avant même que je débute ce doctorat !), ma belle-famille et mes amis (particulièrement Aurélie qui ne s'est jamais lassée de m'encourager et qui n'a jamais douté de me voir arriver au bout de ce doctorat et Marie qui a joué le rôle de correctrice linguistique à la perfection), qui m'ont soutenu tout au long de cette thèse, aussi bien pendant les années de scolarité, mon congé de maternité et la rédaction. Et pour finir, et non le moindre, mes deux gars, Sylvain pour son soutien et son amour inconditionnel et Éliot pour m'avoir permis de voir et surtout de découvrir la vie sous un autre angle.

TABLE DES MATIÈRES

REMERCIEMENTS	III
LISTE DES FIGURES	VIII
LISTE DES TABLEAUX	X
LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES	XI
RÉSUMÉ	XIII
INTRODUCTION	1
DOMAINE D'APPLICATION	1
CONTEXTE DE RECHERCHE	4
<i>Tendances technologiques</i>	<i>5</i>
<i>Tendances en éducation</i>	<i>7</i>
SUJET DE RECHERCHE	10
<i>État de la question : modélisation de l'apprenant</i>	<i>10</i>
<i>Éléments de problématique</i>	<i>13</i>
<i>Objectifs de recherche</i>	<i>15</i>
ORIGINALITÉ DE NOTRE CONTRIBUTION	15
DÉMARCHE DE RECHERCHE	16
PLAN DE LA THÈSE	20
CHAPITRE I - CADRE CONCEPTUEL	21
1.1. LES CONCEPTS DE MODÈLE ET DE MÉTAMODÈLE	22
1.1.1. <i>Modélisation et modèle</i>	<i>22</i>
1.1.2. <i>Métamodèle</i>	<i>23</i>
1.2. L'APPROCHE PAR COMPÉTENCES	24
1.2.1. <i>Définitions</i>	<i>25</i>
1.2.2. <i>Les compétences selon Paquette</i>	<i>30</i>
1.3. LES DIFFÉRENTS ACTEURS INTERVENANT AU COURS DU PROCESSUS D'APPRENTISSAGE (LA GESTION DE POINTS DE VUE MULTIPLES)	34
1.4. CONCLUSION	37
CHAPITRE II - CONTEXTE TECHNOLOGIQUE : ÉTAT DE LA QUESTION	39
2.1. <i>EPORTFOLIO</i>	<i>39</i>
2.1.1. <i>L'outil OSP</i>	<i>43</i>
2.2. LE CONCEPT DE CONTRAT	44
2.3. LA GESTION DE VERSIONS	46
2.3.1. <i>Étude, analyse et comparaison de logiciels de gestion de versions</i>	<i>49</i>
2.4. CONCLUSION	59

CHAPITRE III - PROPOSITION ORIGINALE : UN MODÈLE DE L'APPRENANT COGNITIF, À POINTS DE VUE MULTIPLES ET ÉVOLUTIF	61
3.1. NOTRE PROPOSITION : MODÈLE CONCEPTUEL	62
3.1.1. <i>Modèle cognitif de l'apprenant</i>	62
3.1.2. <i>Modèle de l'apprenant à points de vue multiples</i>	72
3.1.3. <i>Modèle de l'apprenant évolutif</i>	77
3.2. NOTRE PROPOSITION : OPÉRATIONNALISATION	82
3.2.1. <i>Métamodèle, modèle et instance : structure en 3 couches</i>	83
3.2.2. <i>Modèle contexte</i>	84
3.2.3. <i>Modèle de l'apprenant pour un individu particulier (instance)</i>	89
3.2.4. <i>Dynamique d'utilisation</i>	91
3.2.5. <i>Architecture globale</i>	94
3.3. CONCLUSION	96
CHAPITRE IV - IMPLÉMENTATION D'UN PROTOTYPE : NOSMA	98
4.1. MÉTHODE	98
4.2. CONCEPTION DE NOSMA	99
4.2.1. <i>Les acteurs</i>	100
4.2.2. <i>Les cas d'utilisation</i>	100
4.2.3. <i>Le diagramme de classes</i>	106
4.2.4. <i>Les stratégies de négociation conçues et implémentées</i>	109
4.2.5. <i>Les diagrammes de séquence</i>	111
4.3. IMPLÉMENTATION DE NOSMA	115
4.3.1. <i>Outils utilisés</i>	116
4.3.2. <i>Implémentation et fonctionnalités</i>	121
4.3.3. <i>Implémentation d'un lien avec un SAL : TELOS</i>	126
4.4. CONCLUSION	132
CHAPITRE V - ÉVALUATION ET VALIDATION	134
5.1. CHOIX D'UNE MÉTHODE D'ÉVALUATION	134
5.1.1. <i>Méthode d'évaluation</i>	135
5.1.2. <i>Type d'évaluation</i>	136
5.1.3. <i>Critères d'évaluation</i>	137
5.2. DÉMARCHE	139
5.3. SIMULATION 1 : COURS EN LIGNE AVEC ACTEURS HUMAINS	140
5.3.1. <i>Contexte</i>	141
5.3.2. <i>Scénario</i>	143
5.3.3. <i>Simulation</i>	144
5.4. SIMULATION 2 : ÉLABORATION D'UN BILAN DE COMPÉTENCES	147
5.4.1. <i>Contexte</i>	147
5.4.2. <i>Scénario</i>	149
5.4.3. <i>Simulation</i>	150
5.5. SIMULATION 3 : SYSTÈME D'APPRENTISSAGE EN LIGNE	151
5.5.1. <i>Contexte</i>	152
5.5.2. <i>Scénario</i>	154
5.5.3. <i>Simulation</i>	156

5.6. CONCLUSION.....	156
CONCLUSION.....	159
APPENDICE. A - LÉGENDE DU LOGICIEL MOTPLUS ONTO.....	167
A.1 TYPE D'OBJET DU MODÈLE ONTOLOGIQUE.....	167
A.2 DESCRIPTION DES LIENS DU MODÈLE ONTOLOGIQUE UTILISÉS.....	168
APPENDICE. B - ANALYSE DE DÉFINITIONS D'EPORTFOLIO	171
APPENDICE. C - ANALYSE D'OSP	174
C.1 QU'EST-CE QU'OSP	174
C.2 PRINCIPALES FONCTIONNALITÉS D'OSP	175
APPENDICE. D - ANALYSE DES SYSTÈMES DE GESTION DE VERSIONS	181
APPENDICE. E - ANALYSE DE LIBRE SOURCE	206
APPENDICE. F - MODÉLISATION PAR OBJETS TYPÉS (LÉGENDE DU LOGICIEL MOT).....	210
F.1 TYPE D'OBJET MOT STANDARD.....	210
F.2 TYPE DES LIENS ENTRE OBJET MOT STANDARD.....	211
APPENDICE. G - FICHIER DE MAPPING DE CASTOR	213
APPENDICE. H - FICHIER XSL	220
APPENDICE. I - DÉTAILS DES SIMULATIONS	225
I.1. SIMULATION 1 : COURS EN LIGNE AVEC ACTEURS HUMAINS.....	225
I.1.1 Contexte.....	226
I.1.2 Scénario	230
I.1.3 Simulation.....	233
I.2. SIMULATION 2 : ÉLABORATION D'UN BILAN DE COMPÉTENCES.....	245
I.2.1. Contexte.....	245
I.2.2. Scénario	250
I.2.3. Simulation.....	253
I.3. SIMULATION 3 : SYSTÈME D'APPRENTISSAGE EN LIGNE	258
I.3.1. Contexte.....	259
I.3.2. Scénario	262
I.3.3. Simulation.....	265
GLOSSAIRE / LEXIQUE	272
LISTE DE RÉFÉRENCES.....	273

LISTE DES FIGURES

FIGURE 0-I-1. MÉTHODE DE RECHERCHE	17
FIGURE I-1 LE CONCEPT DE COMPÉTENCE.....	27
FIGURE I-2. ONTOLOGIE DU CONCEPT DE COMPÉTENCE (TIRÉ DE PAQUETTE, 2007)	31
FIGURE I-3. EXTENSION DE L'ONTOLOGIE DE COMPÉTENCE AUX HABILETÉS GÉNÉRIQUES (TIRÉ DE PAQUETTE, 2007)	32
FIGURE I-4. EXTENSION DE L'ONTOLOGIE DE COMPÉTENCE AUX INDICATEURS DE PERFORMANCE (TIRÉ DE PAQUETTE, 2007)	33
FIGURE III-1. LE CONCEPT DE COMPÉTENCE DE PAQUETTE (2007) ADAPTÉ.....	63
FIGURE III-2. MODÈLE CONCEPTUEL DU MODÈLE COGNITIF DE L'APPRENANT – ÉTAPE 1	65
FIGURE III-3. MODÈLE CONCEPTUEL DU MODÈLE COGNITIF DE L'APPRENANT – ÉTAPE 2	69
FIGURE III-4. MODÈLE CONCEPTUEL DU MODÈLE COGNITIF DE L'APPRENANT.....	71
FIGURE III-5. MODÈLE DE L'APPRENANT ET SON CONTEXTE SOCIAL	73
FIGURE III-6. MODÈLE DE L'APPRENANT À POINTS DE VUE MULTIPLES	75
FIGURE III-7. EXEMPLE D'UN COURS ET DES ÉVÈNEMENTS QUI PEUVENT GÉNÉRER UNE ÉVOLUTION DU MODÈLE DE L'APPRENANT	80
FIGURE III-8. NOTRE PROPOSITION DE MODÈLE DE L'APPRENANT COGNITIF, À POINTS DE VUE MULTIPLES ET ÉVOLUTIF.....	82
FIGURE III-9. MÉTAMODÈLE, MODÈLE ET INSTANCE.....	84
FIGURE III-10. MODÈLES DE L'APPRENANT PRINCIPAL ET CONTEXTE TEMPORAIRES	89
FIGURE III-11. PROCESSUS <i>DÉBUTER UNE FORMATION</i>	92
FIGURE III-12. PROCESSUS <i>TERMINER UNE FORMATION</i>	93
FIGURE III-13. ARCHITECTURE GLOBALE	95
FIGURE III-14. LES MODÈLES ET LEURS INTERACTIONS	97
FIGURE IV-1. DIAGRAMME DES CAS D'UTILISATION PRINCIPAUX	101
FIGURE IV-2. CAS D'UTILISATION CONSULTER LE MODÈLE DE L'APPRENANT	103
FIGURE IV-3. CAS D'UTILISATION MODIFIER LE MODÈLE DE L'APPRENANT	104
FIGURE IV-4. CAS D'UTILISATION AJOUTER UNE INFORMATION DANS LE MODÈLE DE L'APPRENANT..	105
FIGURE IV-5. CAS D'UTILISATION METTRE À JOUR UNE INFORMATION DU MODÈLE DE L'APPRENANT	106
FIGURE IV-6. DIAGRAMME DE CLASSES.....	108
FIGURE IV-7. DIAGRAMME DE SÉQUENCE AJOUTER UNE PRODUCTION.....	111
FIGURE IV-8. DIAGRAMME DE SÉQUENCE AJOUTER/ÉVALUER UNE COMPÉTENCE	112

FIGURE IV-9. DIAGRAMME DE SÉQUENCE VÉRIFIER UNE COMPÉTENCE.....	113
FIGURE IV-10. DIAGRAMME DE SÉQUENCE ILLUSTRER UNE COMPÉTENCE PAR UNE PRODUCTION.....	114
FIGURE IV-11. DIAGRAMME DE SÉQUENCE ACCRÉDITER UNE PRODUCTION.....	115
FIGURE IV-12. ARCHITECTURE DE SUBVERSION (D'APRÈS COLLINS-SUSSMAN <i>ET AL.</i> 2002)	119
FIGURE IV-13. MODÈLE PROCÉDURAL.....	121
FIGURE IV-14. MODÈLE STRUCTUREL	123
FIGURE IV-15. INTERACTION ENTRE TELOS ET LES COMPOSANTS EXTERNES (TIRÉ DE MIARA, 2010)	127
FIGURE IV-16. FONCTION AJOUTER PRODUCTION DE TELOS	130
FIGURE IV-17. SOUS-NIVEAU DE LA FONCTION AJOUTER PRODUCTION DE TELOS.....	131
FIGURE V-1. MODÈLE DE L'APPRENANT SIMULATION 1	146
FIGURE V-2. CHOIX DE L'ÉVALUATION POUR C1	151
FIGURE V-3. SCÉNARIO PÉDAGOGIQUE SIMULATION 3.....	155

LISTE DES TABLEAUX

TABLEAU I-1. DES MOBILISATIONS EN CASCADE (D'APRÈS JONNAERT, 2002)	26
TABLEAU I-2. PROGRESSION DANS LA PERFORMANCE DE L'EXERCICE D'UNE COMPÉTENCE (D'APRÈS PAQUETTE, 2002A).....	34
TABLEAU II-1 CRITÈRES D'ANALYSE DES SYSTÈMES DE GESTION DE VERSIONS – CATÉGORIE INFORMATION GÉNÉRALE.....	51
TABLEAU II-2 CRITÈRES D'ANALYSE DES SYSTÈMES DE GESTION DE VERSIONS – CATÉGORIE INFORMATION TECHNIQUE	51
TABLEAU II-3 CRITÈRES D'ANALYSE DES SYSTÈMES DE GESTION DE VERSIONS – CATÉGORIE FONCTIONNALITÉS	51
TABLEAU II-4 RÉCAPITULATIF DES CRITÈRES ET DE LEUR VALEUR.....	57
TABLEAU II-5 COMPARATIF DES OUTILS DE GESTION DE VERSIONS	58
TABLEAU III-1. EXEMPLE DE DROITS DES ACTEURS POUR UN CONTRAT PARTICULIER	87
TABLEAU V-1. RÉFÉRENTIEL DE NIVEAUX DU CONTEXTE 2 (MÉQ).....	149

LISTE DES ABRÉVIATIONS, DES SIGLES ET DES ACRONYMES

API : *Application programming interface*

CIG : *Common interest groups*

CV : *Curriculum vitae*

CVS : *Current version system*

DC : *Dublin core*

DOM : *Document object model*

EIAH : Environnement informatique d'apprentissage humain

GDT : Grand dictionnaire terminologique de l'Office québécois de la langue française

HA : hypermédias adaptatifs

IA : Intelligence artificielle

IPP : Informations personnelles et professionnelles

ITS : *Intelligent tutoring system*, systèmes tutoriels intelligents, tuteurs intelligents

LOM : *Learning object metadata*

MDA : *Model driven architecture*

MDD : *Model driven development*

MÉQ : Ministère de l'éducation, du loisir et du sport du Québec

OSP : *Open source portfolio*

PDV : Point de vue

RDCEO : *Reusable definition of competency or educational objective*

SAL : Système d'apprentissage en ligne

SAX : *Simple API for XML*

SGV : Système de gestion de versions

SPI : Situation pédagogique informatisée

SVN : *Subversion*

TIC : Technologies de l'information et de la communication

XML : *eXtensible Markup Language*

RÉSUMÉ

Cette thèse se préoccupe de la modélisation de l'apprenant dans le domaine des Environnements informatiques pour l'apprentissage humain (EIAH). Plus spécifiquement, notre objectif est de proposer un modèle de l'apprenant adapté aux caractéristiques des EIAH développés aujourd'hui. Principalement, les EIAH contemporains visent à être distribués, ouverts, interopérables, développés avec une vision d'apprentissage tout au long de la vie, ancrés sémantiquement et multi-acteurs.

Notre but est ainsi de concevoir et d'opérationnaliser un modèle de l'apprenant qui s'adapte et répond aux besoins des EIAH afin de favoriser la personnalisation des apprentissages. Plus spécifiquement, cette thèse vise la création d'un modèle de l'apprenant offrant des possibilités d'adaptation sémantique des apprentissages, des possibilités d'interopérabilité, une vision de l'apprentissage tout au long de la vie par un suivi de l'évolution dans le temps des apprentissages, une ouverture aux différents contextes de vie d'un individu (professionnel, personnel et académique) et une ouverture aux différents acteurs participant au processus d'apprentissage.

Cet objectif a été atteint par la conception d'un modèle de l'apprenant comportant trois caractéristiques principales, soit un modèle cognitif fondé sur les compétences de l'apprenant permettant un référencement sémantique, un modèle évolutif en lien avec une vision de l'apprentissage comme ayant lieu tout au long de la vie et un modèle multi points de vue permettant la prise en compte des divers acteurs participant au processus d'apprentissage. De plus, notre modèle favorise l'interopérabilité car il utilise le langage XML.

Notre proposition de modèle conceptuel de modèle de l'apprenant est complétée par la conception et l'implémentation d'un système de modélisation de l'apprenant que nous avons appelé Nosma. Ce système est validé par la conduite de trois simulations effectuées dans trois contextes différents d'apprentissage.

Le principal apport de cette thèse est de proposer un cadre pour la modélisation de l'apprenant regroupant plusieurs dimensions : une dimension sémantique (compétences), une dimension évolutive (dans le sens d'apprentissage tout au long de la vie) et une dimension apprentissage ouvert (aux acteurs et aux contextes).

INTRODUCTION

Cette thèse traite de la modélisation de l'apprenant dans le contexte des Environnements Informatiques d'Apprentissage Humain (EIAH). Nous allons présenter dans cette introduction le domaine d'application dans lequel s'inscrit cette thèse, soit l'EIAH. Puis nous exposerons le contexte de notre recherche. Nous poursuivrons en présentant notre sujet de recherche. D'abord, nous ferons un bref état de l'art sur la modélisation de l'apprenant, puis nous présenterons des éléments de problématique qui en découlent. Ensuite, nous énoncerons nos objectifs de recherche, ce qui nous permettra de préciser les contributions originales de la thèse. Finalement, nous exposerons notre démarche ainsi que le plan de la thèse.

Domaine d'application

Cette thèse s'inscrit dans le domaine de l'informatique cognitive. Ce domaine est-il multidisciplinaire (se fondant sur les théories et méthodes de diverses disciplines) ou interdisciplinaire (se trouvant à l'intersection de deux disciplines) ? Il serait multidisciplinaire, comme le laisse suggérer cette définition :

« CI has been described as a new discipline that studies the natural intelligence and internal information processing mechanisms of the brain, as well as processes involved in perception and cognition. CI is a new frontier across disciplines of computing, software engineering, cognitive sciences, neuropsychology, brain sciences, and philosophy. » (Wang, 2007, p. 25)

Mais aussi interdisciplinaire comme le stipule celle-ci :

« Cognitive Informatics (CI) is a transdisciplinary enquiry of cognitive and information sciences that investigates the internal information processing mechanisms and processes of the brain and natural intelligence, and their engineering applications via an interdisciplinary approach. » (Wang, 2010, p. xxii)

Finalement, nous comprenons l'informatique cognitive comme se situant à l'intersection de l'informatique et des sciences cognitives. Les sciences cognitives étant multidisciplinaires, l'informatique cognitive hérite de cette caractéristique. Notre projet se préoccupe plus particulièrement de la modélisation des connaissances (sciences cognitives) et de la création d'un modèle informatique (informatique). Bien que nos problématiques soient issues de l'informatique cognitive, les retombées de notre recherche s'appliquent aux Environnements Informatiques d'Apprentissage Humains (EIAH). Mais qu'est-ce qu'un EIAH ? Avant d'en donner une définition et pour mieux comprendre celle-ci, nous allons présenter un bref historique de l'utilisation de l'ordinateur en éducation. Ce qui suit est une synthèse de différents historiques plus ou moins détaillés (Baron, 1994; Balacheff, Baron, Desmoulins, Grandbastien et Vivet, 1997; Bruillard, 1997; Bruillard *et al.*, 2000; Tchounikine, 2002).

Dans les années 60, des logiciels très simples aussi bien sur le plan éducatif que technique voient le jour. Ce sont des didacticiels aussi appelés Enseignement Assisté par Ordinateur (EAO). Ils permettent de consulter du contenu puis de passer des tests du type question-réponse (questionnaire à choix multiples par exemple). Les rétroactions sont très pauvres, elles fournissent généralement une correction basique du type vrai ou faux.

Dans les années 70, les techniques d'Intelligence Artificielle (IA) commencent à être utilisées pour construire des Systèmes Tutoriels Intelligents (*Intelligent tutoring system*, ITS) ou des systèmes d'Enseignement Intelligemment Assisté par Ordinateur (EIAO). Ces systèmes intègrent une représentation des connaissances du domaine d'apprentissage, une représentation des connaissances de l'apprenant, des mécanismes de raisonnement et des stratégies d'apprentissage. Ces éléments permettent au système de s'adapter à chaque situation. Ainsi, les rétroactions ne sont pas les mêmes : même si elles sont définies à l'avance, elles varient selon l'état des connaissances de l'apprenant et la stratégie pédagogique.

En parallèle, certaines recherches en psychologie mettent en avant l'intérêt de construire par soi-même son apprentissage (apprentissage constructiviste) pour acquérir de nouvelles connaissances mais aussi développer des aptitudes. Les systèmes issus de ce courant de pensée sont les micromondes. Ces environnements sont plus interactifs et permettent davantage d'exploration de la part de l'apprenant. Le micromonde a pour fonction d'utiliser les connaissances de l'apprenant sur le fonctionnement d'objets réels et, à partir de là, lui permettre de les transposer et les formaliser afin d'établir un lien sémantique entre le réel et le formel. Le micromonde le plus connu est LOGO développé par Papert (Papert, 1999).

Dans les années 90, EIAO se transforme pour signifier Environnement Interactif d'Apprentissage avec Ordinateur. Cette nouvelle signification met l'accent sur l'interactivité des systèmes. Les systèmes ne doivent plus simplement transférer des connaissances, ils doivent permettre de les construire à travers les interactions avec l'apprenant. De plus, enseignement devient environnement. Le système n'est pas auto-suffisant, il se doit d'être utilisé dans un environnement (d'enseignement ou de formation, social...).

Finalement, nous parlons dans les milieux francophones d'Environnement Informatique d'Apprentissage Humain (EIAH). Ce changement de sigle dénote une évolution de la vision portée par les recherches utilisant l'ordinateur en éducation. Les principales évolutions entre EIAO et EIAH sont la prise en compte de la généralisation d'internet, de l'évolution de l'informatique distribuée, de l'évolution des méthodes d'apprentissage en collaboration (en éducation) et de l'accroissement de l'offre de formation en ligne. Dans les milieux anglophones, les recherches présentées dans les conférences *Intelligent Tutoring System* (ITS) et *Artificial Intelligence and Education* (AIED) montrent l'évolution et la convergence des recherches sur les tuteurs intelligents, les hypermédias adaptatifs et les environnements d'apprentissage en ligne (Brusilovsky, 2000). Ces recherches ainsi que celles présentées dans les conférences *International Conference on Technology-*

Enhanced Learning (ICTEL) et *European Conference on Technology-Enhanced Learning* (EC-TEL) rejoignent les recherches francophones faites sur les EIAH.

Les EIAH sont définis par Balacheff, Baron, Desmoulins, Grandbastien & Vivet (1997, p. 316) comme :

« des systèmes coopératifs d'apprentissage qui intègrent comme acteurs des enseignants, ou formateurs, et des apprenants, et qui offrent de bonnes conditions d'interaction à travers les réseaux entre agents humains et agents artificiels, ainsi que de bonnes conditions d'accès à des ressources formatives distribuées, humaines et/ou médiatisées ».

Plus récemment, Tchounikine (2009, p. 38) définit le champ scientifique EIAH comme ayant « pour objet d'étudier les questions scientifiques liées aux Situations Pédagogiques Informatisées (SPI) ». Une SPI est « une situation pédagogique intégrant un ou plusieurs logiciels » (p. 34), où une situation pédagogique est « une situation conçue pour amener un apprenant à développer une activité favorable à l'atteinte d'un ou plusieurs objectifs pédagogiques précis » (p. 30). Un EIAH (en tant que logiciel) est « un logiciel spécifiquement conçu dans le but d'amener un apprenant à développer une activité favorable à l'atteinte des objectifs de la SPI considérée » (Tchounikine, 2009, p. 35). Des exemples d'EIAH sont les tuteurs intelligents, les micromondes (ou environnement de simulation), les hypermédias pour l'apprentissage, les environnements d'apprentissage collaboratif, les plateformes de formation à distance et les environnements d'apprentissage mobile. C'est donc dans ce cadre que se place notre thèse.

Contexte de recherche

Le domaine des EIAH, existant depuis de nombreuses années, a évolué rapidement ces derniers temps. Ces évolutions ont trois principales causes : l'évolution des technologies, l'évolution des usages et l'évolution des théories ou cadres conceptuels (Tchounikine, 2009). Les nouvelles tendances abordées en recherche influencent grandement l'évolution des EIAH et amènent de

nouvelles caractéristiques. Nous présentons ces tendances en deux volets, d'abord les tendances technologiques puis les tendances en éducation.

Tendances technologiques

Nous retenons et présentons deux tendances technologiques : les efforts de standardisation et l'émergence du Web sémantique.

La standardisation

Les travaux en technologie de l'information pour l'apprentissage en ligne sont soumis à un effort de standardisation, le but principal étant de permettre l'interopérabilité des systèmes et ainsi la réutilisation du matériel (Pernin, 2006; Devedzic, Jovanovic et Gasevic, 2007). Devedzic (2006) précise que la standardisation permettra aussi un contrôle de qualité (qui se fera ainsi de façon unifiée). Les efforts de standardisation concernent différents aspects des EIAH : les contenus (*Learning Object Metadata*, LOM d'IEEE, 2002; *Sharable Content Object Reference Model* (SCORM)), la conception pédagogique (*Learning design* IMS-LD, 2003), les évaluations (*Question and Test Interoperability*, IMS-QTI, 2006), les acteurs du processus d'apprentissage et leurs connaissances et compétences (*Learner Information Packaging*, IMS-LIP, 2001; *ePortfolio*, IMS-EP 2005; *Reusable Definition of Competency or Educational Objective*, IMS-RDCEO, 2002).

Ces efforts de standardisation font des EIAH actuels des systèmes distribués, ouverts et interopérables. Les EIAH d'aujourd'hui sont conçus afin d'être utilisables ensemble (ou du moins de pouvoir s'interfacer) et d'être utilisables par le plus grand nombre et de partout, pour en maximiser la réutilisation. Bref, les nouveaux EIAH peuvent (ou doivent pouvoir) s'échanger des informations (sur leur contenu, sur leurs utilisateurs...) (Aroyo et Dicheva, 2004). C'est pourquoi ils sont distribués, ouverts et interopérables.

Le Web sémantique

En 2001, Berners-Lee, Hendler et Lasilla (2001) introduisent la notion de web sémantique. Mais qu'est-ce que la sémantique ? En linguistique, sémantique signifie : « Ensemble des relations entre les caractères, ou groupes de caractères, et leur signification, indépendamment de la façon de les employer ou de les interpréter » (Grand dictionnaire terminologique (GDT) de l'Office québécois de la langue française). La sémantique est donc la signification d'un mot, d'un objet, d'une idée ou même d'une situation. Le web sémantique nous offre des méthodes, techniques et outils dont les objectifs sont : (1) d'exprimer formellement la signification et le contenu des ressources; (2) de permettre une recherche « intelligente » des ressources en fonction des requêtes des agents humains ou informatiques; (3) d'effectuer des raisonnements sur et avec ces ressources; (4) de faciliter l'interopérabilité sémantique entre systèmes et la communication entre humains. Selon Anderson et Whitelock (2004), ces objectifs supportent la mise à distance des systèmes d'apprentissages distribués et multi-acteurs, ce web semble ainsi être bénéfique aux EIAH. D'ailleurs Hotte et Contamines (2007) discutent des impacts du web sémantique et des banques de ressources éducatives dans le contexte des communautés d'apprentissage et de formation en ligne.

Les technologies du web sémantique sont de plus en plus utilisées pour les nouveaux EIAH (Aroyo et Dicheva, 2004). Selon Devedzic (2006), récemment, plusieurs projets de recherche concernant des systèmes d'apprentissage fondés sur le web sémantique intègrent des ontologies (ce qui permet un référencement sémantique) et des services du web sémantique pour permettre une personnalisation du processus d'apprentissage. Les ontologies sont utilisées pour référencer les domaines de connaissances des EIAH mais aussi les acteurs, les dimensions techniques, les éléments pédagogiques et même administratifs. Cette tendance technologique, que représente le Web sémantique, permet aux EIAH actuels d'être ancrés sémantiquement.

Tendances en éducation

Les pratiques actuelles en éducation vont, elles aussi, influencer le champ scientifique des EIAH. Nous présentons trois tendances : l'approche par compétences, l'apprentissage tout au long de la vie et la prise en compte des différents acteurs intervenant au cours du processus d'apprentissage.

L'approche par compétences

Parmi les pratiques actuelles en éducation, nous notons particulièrement l'emphasis mise sur les compétences. L'approche par compétences a été adoptée par plusieurs systèmes scolaires nationaux (par exemple, le Ministère de l'Éducation du Québec (MÉQ) en 2000 et la Commission Européenne qui a proposé un référentiel de compétences des TIC en 2008). Cette approche apporte une compréhension renouvelée de l'apprentissage et de l'enseignement. Il s'agit maintenant de permettre aux apprenants de construire des compétences, et non uniquement des connaissances, c'est-à-dire de leur fournir un cadre leur permettant de développer des compétences. Les programmes fondés sur une approche éducative par compétences visent explicitement la construction et le développement de compétences (Scallon, 2004). Il existe différentes définitions du concept de compétence. Paquette (2002b, p. 189) propose celle-ci : « Une compétence est un énoncé de principe qui régit une relation entre un public cible (ou acteur), une habileté et une connaissance. » Nous discuterons plus en détail de ce concept dans le chapitre I de ce document. Roegiers (2000) justifie le rôle fondamental de l'approche par compétences en présentant trois défis que celle-ci devrait permettre de relever : la multiplication des connaissances, qui rend caduque toute pédagogie uniquement fondée sur la transmission des savoirs; la nécessité de plus en plus reconnue de proposer aux élèves des apprentissages significatifs débouchant sur des applications authentiques; et la lutte contre l'échec scolaire, souvent définie en fonction d'éléments plus ou moins pertinents pour la formation des individus. Cette approche nous semble très intéressante car elle offre

un cadre référentiel plus riche que celui de l'approche par objectifs. En effet, avec la compétence, nous savons ce qu'une personne sait (connaissance) et à quel degré elle maîtrise cette connaissance dans un usage en contexte (habileté).

L'apprentissage tout au long de la vie

L'apprentissage est un processus ayant lieu tout au long de la vie (notamment avec l'expansion de l'apprentissage continu), se déroulant dans des contextes variés et faisant intervenir différents acteurs. La société du savoir nécessite un apprentissage continu. En effet, cette société du savoir nécessite une adaptabilité et une flexibilité accrue; pour l'atteindre, les acteurs de cette société doivent continuellement se former pour rester compétitifs (Prax, 2003). Adopter une vision d'apprentissage tout au long de la vie constitue un grand défi actuel dans le monde de l'éducation. Les EIAH d'aujourd'hui sont conçus dans cette optique. Le processus d'apprentissage peut s'ouvrir aux différents contextes de son occurrence (contexte scolaire, professionnel, personnel, formation continue ou non...) et aux différents acteurs y participant (l'apprenant lui-même mais aussi des professeurs, des tuteurs, des collègues apprenants et des acteurs machines dans le cadre d'apprentissage en ligne). De plus, l'apprentissage est un processus évolutif et cette évolution peut elle-même être source d'apprentissage, tout du moins, disposer de cette évolution permet d'effectuer un retour réflexif sur ses apprentissages ce qui peut être bénéfique pour les apprenants (Brown, Collins et Duguid, 1989). Il est ainsi primordial que les EIAH abordent l'apprentissage sans oublier ses aspects sociaux et évolutifs ni les différents contextes de son occurrence.

Prise en compte des différents acteurs

Les EIAH sont conçus pour être utilisés par différents types d'acteurs: les professeurs, les apprenants, les membres du personnel administratif, soit les différents acteurs intervenant au cours du processus d'apprentissage (Aroyo et Dicheva, 2004).

Ces différents acteurs peuvent utiliser un EIAH consécutivement, en parallèle ou en collaboration. Les EIAH d'aujourd'hui sont utilisés tout au long du cycle de vie d'une formation, soit de sa conception pour un concepteur pédagogique ou un professeur à son utilisation par un apprenant en passant par sa production, son évaluation, son administration... C'est pourquoi ces EIAH sont multi-acteurs. Ces différents intervenants effectueront des tâches différentes selon leurs besoins.

Il existe d'autres approches éducatives influençant la conception et la construction d'EIAH, comme l'apprentissage situé (Lave et Wenger, 1990), l'apprentissage collaboratif (Brown, Collins, et Duguid, 1989), les communautés de pratique et d'apprentissage (Barab et Duffy, 2000).

Les tendances actuelles, aussi bien en éducation que technologiques, rendent les EIAH d'aujourd'hui distribués, ouverts, interopérables, développés avec une vision d'apprentissage tout au long de la vie, ancrés sémantiquement et multi-acteurs. Ces caractéristiques regroupées permettent de bénéficier d'une infrastructure permettant à des agents « intelligents » d'effectuer des raisonnements afin d'offrir des fonctionnalités élaborées de scénarisation pédagogique, d'adaptation et de personnalisation (Aroyo et Dicheva, 2004; Devedzic, 2006). C'est dans ce contexte d'EIAH que se situe notre recherche. Plus précisément, nous abordons le domaine des EIAH en nous centrant sur l'apprenant en situation de formation tout au long de la vie, comme proposé par les chercheurs introduisant les Environnements personnels d'apprentissages (ou *Personal Learning Environments*, PLE). Les PLE proposent un focus différent lors de la conception d'EIAH afin de supporter l'apprentissage tout au long de la vie, l'apprentissage formel et informel et la personnalisation (Wilson *et al.* 2006). Notre recherche se rapproche de cette branche, puisque nous appréhendons les EIAH avec cette vision d'apprentissage tout au long de la vie et dans des contextes formels et informels. Nous continuerons toutefois d'employer le terme EIAH dans ce document.

Sujet de recherche

Nous nous intéressons à la modélisation de l'apprenant dans le cadre des EIAH actuels. Dans la documentation scientifique, différents termes sont utilisés pour désigner la modélisation de l'apprenant. En ce qui concerne l'apprenant nous trouvons: utilisateur, apprenant ou étudiant et pour modélisation nous trouvons profil ou modèle. Nous allons adopter les termes « modèle de l'apprenant » : apprenant parce que nous nous intéressons aux utilisateurs utilisant un système d'apprentissage pour apprendre (aussi bien des étudiants dans une situation d'apprentissage scolaire que des adultes vivant des situations d'apprentissage dans leurs environnements de travail), et modèle parce que notre approche consiste à définir un modèle générique qui peut contenir différents profils d'apprenants. Nous entendons par modèle de l'apprenant une structure générique contenant certaines informations sur l'apprenant.

Dans la suite de cette section, nous allons présenter un bref état de la question pour en tirer des éléments de problématique. Nous finirons en présentant notre objectif de recherche.

État de la question : modélisation de l'apprenant

Qu'est-ce qu'un **modèle de l'apprenant** dans le cadre d'EIAH ? Quelles informations contient-il ? Dans les écrits en sciences de l'éducation, différents aspects ont été identifiés lors d'une activité d'apprentissage :

- (1) le psychomoteur (réfère à la capacité psychomotrice, à la coordination et aux mouvements physiques);
- (2) l'affectif (fait référence aux émotions, aux sentiments et aux attitudes de l'apprenant);
- (3) le cognitif (recouvre les capacités intellectuelles, porte sur les connaissances dans un domaine ainsi que sur les processus de traitement de l'information);

- (4) le métacognitif (contient deux dimensions : les métaconnaissances (ou connaissances sur les connaissances) et les stratégies de gestion de l'apprentissage).

Ces quatre aspects doivent tous être pris en compte lors de la conception d'activités d'apprentissage. Le but des modèles de l'apprenant est de fournir au système d'apprentissage des informations pertinentes pour personnaliser, adapter l'apprentissage à des caractéristiques ou préférences de l'apprenant. Historiquement, les modèles de l'apprenant sont des modèles cognitifs, c'est-à-dire représentant les connaissances de l'apprenant sur un domaine particulier (celui sur lequel porte l'environnement d'apprentissage). Sison et Shimura (1998) décrivent trois approches différentes pour concevoir un modèle de l'apprenant : **l'approche par recouvrement** qui consiste à comparer la connaissance de l'apprenant à celle de l'expert pour estimer la part de connaissance qui reste à acquérir ; **l'approche analytique** par laquelle l'observation d'un comportement de l'apprenant, différent du comportement attendu, permet d'isoler un problème (ou *bug*) et **l'approche synthétique** qui généralise le modèle de l'élève à partir de l'observation du comportement de plusieurs apprenants.

Les modèles d'apprenant ont été particulièrement utilisés dans les systèmes tutoriels intelligents (ITS). Ces systèmes ont surtout été développés dans les années 80. Leur singularité est qu'ils sont fortement liés aux recherches sur les systèmes à base de connaissance développés en intelligence artificielle (IA). La plupart des auteurs proposent la même architecture générale pour les ITS. Elle est composée de quatre modules : le modèle du domaine ou de l'expert (représentant les connaissances du domaine), le modèle de l'apprenant (représentant les connaissances de l'apprenant), le modèle pédagogique (représentant le ou les scénarios pédagogiques) et l'interface (Wenger, 1987; Self, 1990; Nkambou, Bourdeau et Mizoguchi, 2010).

Le modèle de l'apprenant est orienté domaine car il contient les connaissances de l'apprenant sur le domaine concerné par l'ITS (par exemple l'arithmétique ou alors les faits historiques de la seconde guerre mondiale). Le but du modèle de l'apprenant

est en quelque sorte de construire un portrait des connaissances de l'apprenant qui s'enrichit à chaque étape de l'apprentissage. Cette partie du système a besoin de recueillir des informations qui peuvent être de nature différente (Nicaud et Vivet, 1988). Des informations implicites issues de l'analyse du comportement de l'élève pendant les sessions ainsi que des informations explicites issues du « dialogue » direct entre le système et l'apprenant, permettent, entre autres, l'élaboration de ce modèle de l'apprenant. John Self (1988) définit le modèle de l'apprenant comme un *4-uplet* contenant des connaissances procédurales, des connaissances conceptuelles, des traits individuels et l'historique de l'apprenant. Il explique aussi que les modèles de l'apprenant sont utilisés en ITS pour répondre à des questions que le système se pose sur l'apprenant. En ITS, le modèle de l'apprenant est donc un modèle cognitif qui représente les connaissances de l'apprenant et qui est utilisé pour adapter le système aux besoins particuliers des apprenants.

De plus en plus apparaissent des modèles affectifs portant sur les émotions, les buts des apprenants, etc. Hibou et Py (2006) stipulent que les modèles de l'apprenant contiennent des informations de nature cognitive, comportementale ou psychologique (ce qui rejoint les domaines touchés lors d'activités d'apprentissage). Ces informations peuvent être des connaissances, des savoir-faire, des intentions ou des émotions. Hibou et Py présentent quelques types d'informations pouvant apparaître dans un modèle de l'apprenant : modèle de l'activité (permet de visualiser le parcours de l'apprenant), modèle comportemental (constitué d'un ensemble d'entités comportementales chacune formée d'un type de comportement associé à un coefficient de certitude), modèle des plans et des intentions (dans le cas de tâche de résolution de problème) et modèle des motivations.

En plus d'adapter l'apprentissage, les modèles d'apprenant peuvent aussi servir à organiser et gérer l'interaction de l'apprenant avec le système, à évaluer les connaissances de l'apprenant et à simuler et prédire le comportement de l'apprenant (Hibou et Py, 2006). C'est la nature des informations contenues dans le modèle de l'apprenant qui rend possible de l'utiliser pour tel ou tel but. Si le modèle contient les

connaissances de l'apprenant, il sera alors possible d'évaluer celles-ci, s'il contient les buts d'apprentissage de l'apprenant, il sera possible d'adapter la formation aux besoins de l'apprenant, etc.

Éléments de problématique

Les modèles d'apprenant sont depuis longtemps utilisés en EIAH. La question que nous nous posons est : comment un modèle d'apprenant peut-il répondre aux caractéristiques des EIAH actuels ? Les modèles d'apprenant développés dans les ITS sont des modèles computationnels et sont utilisés pour effectuer un diagnostic sur l'état cognitif de l'apprenant concernant le domaine visé par l'ITS. Dans le cadre de notre projet, nous envisageons aussi un modèle représentant les connaissances de l'apprenant. L'approche par compétences étant une pratique actuelle en éducation, pourquoi ne pas disposer d'un modèle cognitif de l'apprenant représentant ses compétences (et non uniquement ses connaissances) ? Aussi, la tendance visant le développement de systèmes pour l'apprentissage tout au long de la vie nous incite à prendre en compte cet élément dans notre recherche. Ainsi, nous souhaitons un modèle de l'apprenant contenant les connaissances ou compétences de l'apprenant sur plusieurs domaines (et non sur un seul domaine particulier). Cela permettra aussi au modèle de se « brancher » à différents systèmes visant l'apprentissage de différents domaines. Dans le cadre de notre projet de thèse, nous effectuons donc une modélisation cognitive de l'apprenant mais nous n'avons pas pour objectif de créer un modèle cognitif reproduisant la cognition humaine (dans le sens de la copier, la reproduire). Notre but est plutôt de représenter pertinemment et dynamiquement (dans son évolution cognitive) l'apprenant afin de fournir au système d'apprentissage des informations lui permettant de s'adapter aux caractéristiques et besoins de l'apprenant, d'évaluer ses connaissances, etc.

D'autres éléments caractérisant les EIAH actuels sont des facteurs clés à prendre en considération lors de la création d'un modèle de l'apprenant, notamment

le fait que les EIAH soient développés pour des contextes variés (ouverts et interopérables) et qu'ils soient multi-acteurs. Aussi, comme nous l'avons vu, de nouvelles technologies peuvent être utilisées, comme celles du web sémantique par exemple. Finalement, nous ne connaissons pas de modèle de l'apprenant correspondant à toutes ces caractéristiques.

Ainsi, notre problématique concerne la construction d'un modèle de l'apprenant répondant aux caractéristiques des EIAH actuels. Plus spécifiquement, sur le plan cognitif, elle porte sur la constitution d'un modèle cognitif de l'apprenant, c'est-à-dire une modèle représentant l'état cognitif d'un apprenant à un moment donné. Le modèle cognitif de l'apprenant que nous proposons intègre des compétences décrites en termes de connaissance et d'habileté et exemplifiées par des productions de celui-ci. De plus, l'apprentissage est un processus dynamique dont l'historique de l'évolution est un élément important et constitutif du modèle. L'apprentissage est aussi un processus qui sollicite la participation de différents acteurs. Un modèle de l'apprenant doit donc permettre de représenter et d'étudier l'évolution de l'apprentissage dans le temps et le contexte social dans lequel celui-ci a lieu. Sur le plan informatique, notre problématique intègre l'instrumentation de ce modèle par un outil. Particulièrement, le modèle est évolutif (gestion des versions dans le temps) et multi-acteurs (gestion des versions d'acteurs ou de groupe d'acteurs).

Relativement à cette problématique, nous nous posons la question suivante : dans une optique d'apprentissage tout au long de la vie, comment créer un **modèle de l'apprenant** afin d'offrir :

1. une représentation cognitive de l'apprenant, en termes de compétences
2. un suivi de l'évolution dans le temps des apprentissages
3. une ouverture aux différents contextes de vie d'un individu (professionnel, personnel et académique)
4. et une ouverture aux différents acteurs participant au processus d'apprentissage.

Objectifs de recherche

Notre objectif est de proposer un modèle cognitif d'une personne en situation d'apprentissage, en vue de supporter l'évolution de ses apprentissages ayant lieu dans différents contextes, avec différents acteurs et tout au long de sa vie. D'où découle un deuxième objectif consistant à formaliser et opérationnaliser le modèle de l'apprenant.

Originalité de notre contribution

Les recherches en modélisation de l'apprenant datent de nombreuses années. Leur évolution, en parallèle de l'évolution du domaine des EIAH, est un courant de recherche actuel dans lequel s'inscrit notre contribution. L'originalité de notre contribution sur le plan conceptuel est l'intégration dans un même modèle de l'apprenant de diverses approches et techniques issues, notamment, des nouvelles caractéristiques des EIAH et des pratiques actuelles dans le domaine des sciences de l'éducation.

Plus précisément, l'originalité de ce projet de thèse est de prendre en compte différentes dimensions dans un même modèle de l'apprenant : une dimension sémantique (connaissances et compétences), une dimension évolutive (dans le sens d'apprentissage tout au long de la vie) et une dimension d'apprentissage ouvert (aux acteurs, aux contextes).

Sur le plan informatique, notre contribution originale est la formalisation du modèle conceptuel de modèle de l'apprenant et son opérationnalisation en un prototype permettant aux acteurs concernés d'utiliser et de manipuler le modèle de l'apprenant conçu et intégrant des fonctionnalités innovantes répondant aux besoins des EIAH actuels. Par exemple, l'outil permet de gérer l'apprentissage par compétence, de suivre l'évolution des apprentissages, il permet aussi l'implication de différents acteurs et il peut être utilisé dans divers contextes.

Démarche de recherche

Pour atteindre nos objectifs, nous avons développé une démarche de recherche intégrant d'une part la méthode du processus unifié (Larman, 2005) et d'autre part des techniques de recherche-développement, de modélisation et simulation (Van der Maren 1996, 2003). Ceci nous a permis de définir une démarche de recherche itérative, axée sur l'affinement successif de la solution conceptuelle proposée ainsi que du prototype informatique. Nous avons choisi le processus unifié pour quatre raisons principales. Premièrement, tous les travaux de ce projet de thèse vont nourrir l'objectif final (final en termes de séquence et non final en termes de finalité de ce projet de recherche) consistant au développement d'un prototype informatique et justement, le processus unifié est un processus de développement qui décrit les cycles permettant de construire un logiciel. Deuxièmement, le processus unifié est relativement utilisé dans les projets orientés objet (Larman, 2005), aussi bien pour l'analyse que pour la conception orientées objet. Troisièmement, il encourage de bonnes pratiques largement reconnues (Larman, 2005). Quatrièmement, « le processus unifié est souple et très ouvert, et il incite à inclure des pratiques judicieuses issues d'autres méthodes itératives » (Larman, 2005, p. 35). Ceci semble laisser une porte ouverte pour nous permettre d'inclure des méthodes issues des sciences cognitives dans notre démarche.

Notre méthode comporte ainsi trois phases inspirées du processus unifié (tel qu'illustré à la figure suivante), la première étant l'inception, la seconde est celle de l'élaboration et de la construction puis la troisième est celle de transition. Nous présentons ici la méthode dans sa globalité, c'est-à-dire ses trois phases et leurs étapes. Nous allons présenter chacune des étapes plus en détail dans les chapitres traitant et présentant les résultats de celles-ci.

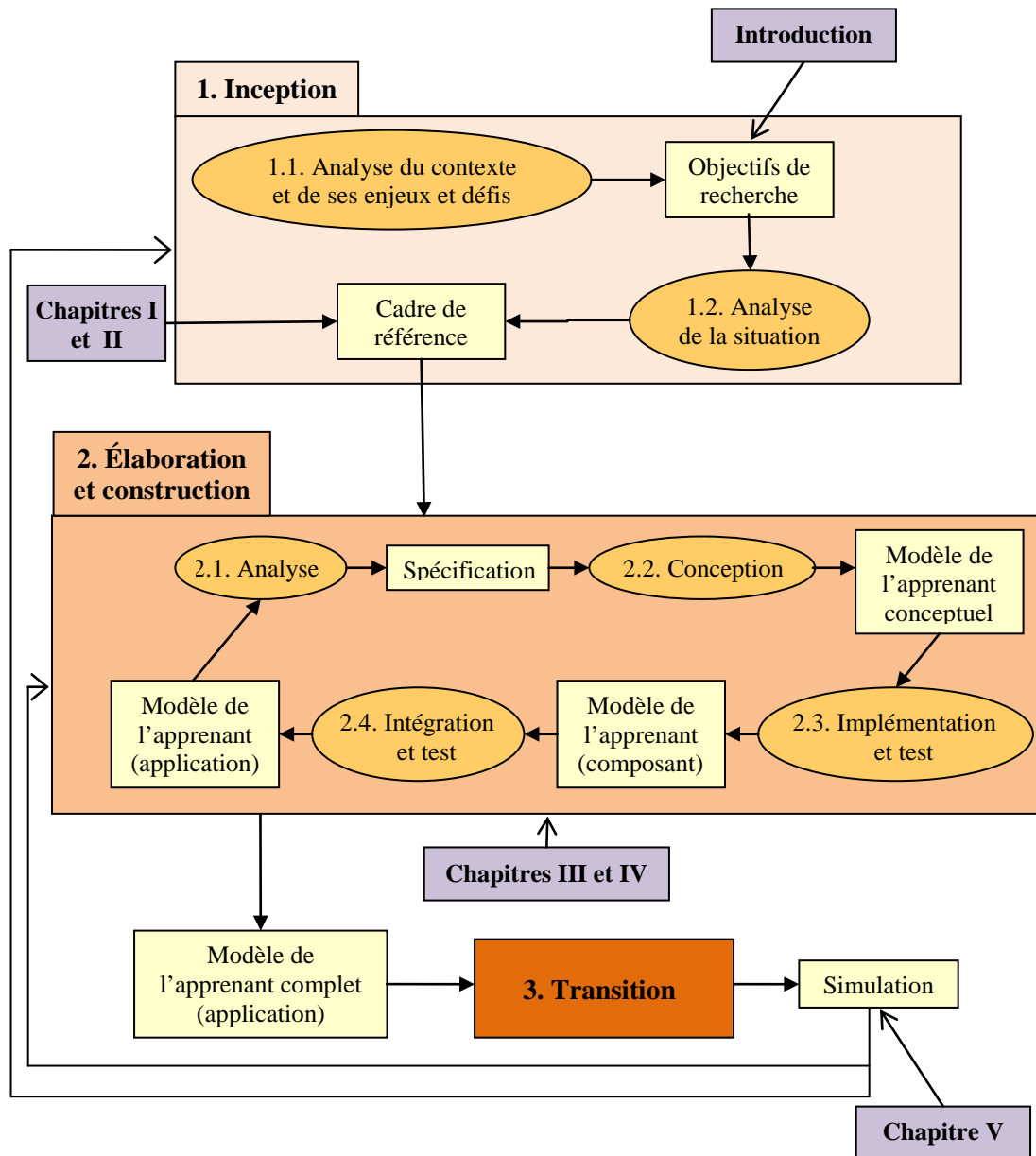


Figure 0-I-1. Méthode de recherche

Phase 1 : Inception : « Elle consiste en une vision approximative de la finalité du projet, une étude d'opportunité, une définition du périmètre et des estimations globales. » (Larman, 2005, p. 40).

Cette première phase du processus unifié consiste à définir une vision globale du projet ainsi que de vérifier sa faisabilité. Habituellement, cette phase se fait en collaboration étroite avec l'utilisateur soit le client pour lequel le système est

développé. Dans notre cas, nous ne disposons pas de client ou d'utilisateur, ce projet étant un projet de recherche. Pour effectuer cette phase, nous nous sommes inspirés de la méthodologie de recherche-développement (Van der Maren, 1996, 2003). La première étape consiste à analyser le contexte de ce projet de thèse, ses enjeux et ses défis. Cela nous permet d'identifier notre problématique et de définir nos objectifs de recherche. Ces étapes et résultats ont été présentés dans les sections précédentes de cette introduction. À l'aide de ces objectifs, nous pouvons entamer la deuxième étape qui consiste à analyser la situation actuelle plus en détail, cette analyse nous donne l'état de la question, elle correspond à la revue de la documentation. Cette analyse se fait sur les différents aspects liés à nos objectifs de recherche. Cette étape est présentée dans les chapitres I et II. Une fois l'inception effectuée, nous disposons d'un cadre de référence avec lequel nous pouvons commencer les phases suivantes.

Phase 2 : Élaboration et construction

Élaboration: « Elle débouche sur une vue plus élaborée, avec implémentation itérative de l'architecture noyau, la résolution des risques élevés, l'identification de la plupart des besoins et du périmètre réel et des estimations plus réalistes. » (Larman, 2005, p. 40).

Construction: « Il s'agit de l'implémentation itérative des éléments qui présentent des risques et une complexité moindres, et de la préparation du déploiement. » (Larman, 2005, p. 40).

Ces deux phases du processus unifié consistent à l'implémentation itérative et incrémentale du projet. Nous avons fusionné ces deux phases car l'ampleur de cette recherche ne justifie pas de les distinguer. Cette deuxième phase de notre méthode se découpe en quatre étapes qui se font de façon itérative. Chaque itération produit un système effectif qui est un sous ensemble du produit final. La première étape est l'analyse. Elle permet de produire des spécifications. Au cours de cette étape, nous avons, par exemple, étudié différents outils existants et à partir desquels nous pouvons travailler. Les spécifications servent de base à la deuxième étape qui est la conception. Cette étape consiste au développement de modèles statiques et de modèles dynamiques. Le produit résultant de cette étape est un modèle de conception

de notre modèle de l'apprenant. L'étape d'implémentation et test consiste à implémenter le modèle de conception sous formes de composants informatiques et de les tester (test des composants séparément). La dernière étape, celle d'intégration et test consiste à intégrer les composants du système et à tester le système développé (ce qui correspond à tester l'interaction entre les différents composants informatiques du système). Le produit de cette étape est une application de modèle de l'apprenant. À chaque itération, un système fonctionnel est créé, il représente un sous-ensemble du système final. À la fin de cette étape, le cycle peut se poursuivre, une nouvelle itération sera alors déclenchée. Après la dernière itération, le modèle de l'apprenant complet est développé et nous pouvons passer à la dernière phase. Nous avons procédé en quatre itérations, comme nous l'expliquerons plus en détail dans le chapitre IV. Les résultats de l'étape de conception sont présentés au chapitre III. Et les résultats des étapes concernant le développement du système sont présentés au chapitre IV.

Phase 3 : Transition : « C'est le moment des bêta tests et du déploiement. » (Larman, 2005, p. 40).

Dans le cadre de ce projet, nous n'avons pas effectué de déploiement. Cette phase consiste donc uniquement en tests. Ces tests ont été effectués sous forme de simulations, à l'aide de scénarios définis selon les cas d'utilisation développés précédemment. Nous avons choisi de tester le système par simulation car il nous serait difficile de disposer de systèmes d'apprentissage en ligne réels sur lesquels lier notre modèle pour tester son fonctionnement. Il serait aussi difficile de trouver des acteurs humains en nombre suffisant pour tester le fonctionnement du modèle. Les résultats de ces simulations sont présentés au chapitre V. À la suite de ces simulations, il est possible de revenir à la phase d'inception ou à la phase d'élaboration. En effet, les résultats obtenus lors des simulations pourront avoir des conséquences sur ces deux phases. Dans le cadre de notre projet de thèse, nous n'effectuons pas de nouvelles itérations, nous soulignons les résultats obtenus lors de

l'analyse des simulations et les éventuels effets qu'ils pourraient avoir sur des améliorations futures.

Plan de la thèse

Dans le chapitre I, nous présentons le cadre conceptuel de cette thèse. Nous étudions d'abord les concepts de modèle et de métamodèle. Ensuite, nous nous préoccupons de l'approche par compétences pour laquelle nous présentons des définitions et l'approche que nous choisissons d'utiliser. Finalement, nous discutons de l'implication de différents acteurs dans le processus d'apprentissage et de la façon de les prendre en compte. Pour conclure ce chapitre, nous précisons ce que nous retenons de chacune de ces approches pour notre thèse.

Le chapitre II est aussi un chapitre d'état de la question mais il traite du contexte technologique en lien avec notre thèse. En premier, nous étudions l'ePortfolio et plus particulièrement l'outil OSP. Ensuite, nous nous attardons sur le concept de contrat qui permet de gérer les interactions d'un système informatique. Finalement, nous étudions la gestion des versions et comparons différents systèmes de gestion de versions existants.

Le chapitre III est le cœur de la thèse et vise la présentation de notre proposition originale de modèle de l'apprenant. Plus précisément, nous présentons le modèle conceptuel de la solution que nous proposons. Nous présentons aussi comment ce modèle s'utilise, soit une vue dynamique du modèle conceptuel.

Le chapitre IV concerne la conception et l'implémentation de notre prototype mettant en œuvre le modèle conceptuel présenté au chapitre III.

Quant au chapitre V, il présente les simulations que nous avons faites sur le prototype qui nous permettent de valider nos objectifs de recherche.

Chapitre I - CADRE CONCEPTUEL

Pour répondre à notre objectif de recherche consistant à proposer un modèle cognitif d'une personne en situation d'apprentissage, en vue de supporter l'évolution de ses apprentissages ayant lieu dans différents contextes, avec différents acteurs et tout au long de sa vie, nous nous sommes intéressés à différents concepts et technologies. Plus particulièrement, nous avons étudié les notions de métamodèle et de modèle. Le concept de modèle est directement lié à notre recherche puisque nous visons la création d'un modèle de l'apprenant. Comme nous l'expliquerons, nous nous intéressons au concept de métamodèle car celui-ci nous permet de clarifier l'exposé de nos idées. Nous avons aussi étudié l'approche par compétence (en tant que technique de modélisation cognitive de l'apprenant) puisque c'est sur celle-ci que notre modèle cognitif de l'apprenant repose. Nous complétons cette approche avec la notion de productions, qui permet de servir d'exemple aux compétences. Concernant la dynamique du modèle, son évolution dans le temps, dans différents contextes et en relation avec différents acteurs, nous nous sommes intéressés à la notion de point de vue, à la notion de contrat et à la gestion des versions. Nous explorons la notion de point de vue comme solution pour gérer et prendre en compte les différents acteurs pouvant interagir avec le modèle. Nous nous sommes intéressée à la notion de contrat et à la gestion de versions parce que nous voulons un modèle qui soit utilisable dans différents contextes et tout au long de la vie.

Dans ce chapitre, nous présentons le cadre conceptuel et nous présenterons dans le chapitre II le contexte technologique de notre recherche. Plus précisément, ce chapitre correspond à la première phase de la méthodologie, soit celle d'inception qui consiste à définir une vision globale du projet et plus précisément à l'étape 1.2, soit

celle d'analyse de la situation. Dans ce chapitre I, nous présentons trois thèmes essentiels à notre recherche. D'abord, nous explorons les concepts de modèle et de métamodèle (1.1). Ensuite, nous nous préoccupons de l'approche par compétences (1.2). Finalement, nous étudions les points de vue et les difficultés de gestion liées à leur utilisation dans un système informatique (1.3.).

1.1. Les concepts de modèle et de métamodèle

1.1.1. Modélisation et modèle

Notre projet consiste à modéliser l'apprenant dans le cadre d'EIAH. Mais qu'est-ce que concevoir un modèle ? Seidwitz (2003, p. 27) définit un modèle comme « *a set of statements about some system under study* ». Tandis que Bézivin et Gerbé (2001, p. 2) propose plutôt: « *a model is a simplification of a system built with an intended goal in mind* ». Un modèle est ainsi une représentation d'un objet ou d'un système réel, représentation de certaines dimensions sans tenir compte d'autres dimensions.

Selon Le Moigne (1990, p. 53), la modélisation est une :

« action d'élaboration et de construction intentionnelle, par composition de symboles, de modèles susceptibles de rendre intelligible un phénomène perçu complexe, et d'amplifier le raisonnement de l'acteur projetant une intention délibérée au sein d'un phénomène; raisonnement visant notamment à anticiper les conséquences de ces projets d'actions possibles ».

Baker (2000) présente trois caractéristiques générales des modèles :

- un modèle est utilisé pour prédire les incidences d'un phénomène dans un champ expérimental déterminé;
- un modèle permet d'élaborer ou de raffiner la théorie sur laquelle il est fondé;
- un modèle contraint nécessairement à l'abstraction de certains phénomènes (ensemble d'objets ou évènements).

Il existe un consensus sur l'existence d'une relation du type « représentation de » entre le modèle et le système étudié (Seidwitz, 2003, Nodenot, 2005), le modèle

représente le système. Un modèle peut être descriptif ou prescriptif. Un modèle descriptif décrit un système existant. Il est considéré comme juste s'il reproduit le comportement du système réel. Généralement, les modèles descriptifs servent à évaluer des théories (Nodenot, 2005). Les modèles prescriptifs servent à la spécification du comportement que devra avoir le système à développer.

Baker (2000) présente trois utilisations différentes de modèles dans le cadre d'environnement informatique pour l'apprentissage :

- le modèle comme outil scientifique (modèle utilisé pour comprendre et prédire certains aspects d'une situation éducative);
- le modèle comme composant d'environnement d'apprentissage (modèle computationnel correspondant à un aspect du processus d'apprentissage et utilisé comme composant de l'environnement, par exemple, le modèle de l'apprenant);
- le modèle comme base pour la conception d'environnement d'apprentissage (modèle de processus éducationnels, et des théories liées).

Selon l'auteur, les modèles utilisés en éducation ne doivent pas nécessairement être symboliques, computationnels et/ou reproduire la cognition humaine. Il pense même que des modèles non computationnels joueront un rôle important dans le futur des environnements informatiques d'apprentissage.

Pour notre projet, nous avons conçu un modèle descriptif représentant une personne en situation d'apprentissage (un apprenant) selon certaines dimensions (ses compétences, ses productions et d'autres éléments présentés par la suite).

1.1.2. Métamodèle

Nous proposons un modèle de l'apprenant utilisable dans différents contextes d'apprentissage. Nous avons choisi la notion de métamodèle pour nous aider à expliquer notre projet. Dans cette section, nous allons présenter la signification conceptuelle du métamodèle. Nous verrons dans le chapitre III comment cette notion est utile à notre recherche.

Alors qu'un modèle est la représentation d'un objet ou d'un système réel, un métamodèle sert à exprimer les concepts communs à l'ensemble des modèles d'un même domaine (Encyclopédie scientifique en ligne). Dans le domaine du développement de logiciels, la tendance va vers une approche orientée modèle (au contraire de l'approche centrée code), où les modèles ont une place privilégiée dans le développement. Dans cette optique, le métamodèle est un modèle d'un langage de modélisation. C'est la définition d'un langage permettant de créer des modèles. « *A meta-model defines the structure, semantics, and constraints for a family of models* » (Mellor, Scott, Uhl et Weise 2004, p. 14). Ou encore, « *a meta-model is a model that defines the language for expressing a model* » (OMG, 2002, p. 3). D'après Terrasse, Savonnet, Leclercq, Grison et Becker (2005), le métamodèle est à la fois un modèle abstrait, la définition d'un langage et le support de mécanismes d'abstraction. La relation liant un métamodèle et un modèle est « conforme à » : un modèle est conforme à un métamodèle (Bézivin et Gerbé, 2001 d'après Nodenot, 2005). Nodenot (2005) utilise le terme « filtre » pour expliquer le métamodèle. Un modèle est une représentation partielle d'un objet ou système (toute la réalité de cet objet ou système n'est pas représentée dans le modèle mais seulement certains de ses attributs). Le modèle résulte donc de l'application d'un filtre sur le monde réel. Ce filtre réside dans le métamodèle qui précise les caractéristiques du monde réel qui peuvent apparaître dans les modèles. C'est dans cette mesure que nous employons le terme de métamodèle dans cette thèse. Nous verrons au chapitre III que l'utilisation de la notion de métamodèle, dans le sens donné précédemment, nous permet d'exprimer plus clairement notre proposition de modèle de l'apprenant.

1.2. L'approche par compétences

Une tendance actuelle dans le monde de l'éducation est l'approche par compétences. Nous souhaitons créer un modèle cognitif de l'apprenant riche

sémantiquement et nous pensons que cette approche de modélisation de l'apprenant par compétences pourrait répondre à ce besoin. Nous allons donc explorer ce concept de façon plus approfondie en présentant plusieurs de ses définitions. Nous terminons en présentant en détail l'approche du concept de compétence que nous retenons dans le cadre de notre recherche, soit celle de Paquette (2002a).

1.2.1. Définitions

Nous trouvons dans la documentation beaucoup de définitions du concept de compétence. Nous en présentons et analysons quelques-unes dans cette section.

Jonnaert (2002) présente des définitions du concept de compétences issues de différents domaines pour montrer le cheminement de ce concept avant d'être utilisé en éducation. Dès le début du 20ème siècle, les **linguistes** ont défini le concept de compétence en le différenciant du concept de performance. La compétence linguistique est un potentiel individuel non encore activé et la performance est l'activation, en situation de communication, de la compétence linguistique. Les **psychologues du développement cognitif** acceptent cette distinction entre compétence et performance, mais ils notent toutefois un décalage entre la compétence (qui est définie a priori) et la performance du sujet en action (qui est réellement observée). C'est la situation qui crée la différence entre la compétence et son actualisation par la performance. Les situations sont ainsi sources des performances et critères des compétences. Depuis une vingtaine d'années, les **spécialistes des sciences du travail** ont eux aussi développé leur approche du concept de compétence. Avant cela, ils parlaient plutôt de qualification, qui est un concept prescriptif, défini a priori. La compétence devient la capacité d'un individu à gérer son potentiel en situation, elle est définie en référence à l'action du sujet en situation. Le contexte est donc pris en compte. Dans cette perspective, les concepts de compétence et performance sont fusionnés. Il n'y a plus de distinction entre prescriptif et observé. En éducation, le décalage existant entre compétence et performance n'est pas acceptable. En effet, le

mandat des enseignants est bien de faire développer des compétences à leurs élèves. Il est attendu qu'à la fin d'une formation, un étudiant maîtrise les compétences pour lesquelles il a été formé. De plus, Raynal et Rieunier (1997, dans Jonnaert 2002) décrivent la problématique suivante : comment, dans un contexte scolaire, différencier la compétence de la performance ? Comment, en éducation, définir une compétence sans préciser comment elle s'actualisera en situation, c'est-à-dire sans nommer une performance ? Compétence et performance sont toutes deux dans la situation qui est le cœur de l'action de l'apprenant. Et comment différencier compétence, performance et situation ? La compétence est autant déterminée par la situation qu'elle la détermine elle-même. Le passage au domaine de l'éducation n'est pas aisé à réaliser. Il existe des approches pédagogiques du concept de compétences qui se sont éloignées des approches présentées précédemment, pour adapter le concept au contexte particulier des situations d'apprentissage. On peut toutefois remarquer un rapprochement avec la perspective développée par les spécialistes des sciences du travail. Il existe de nombreuses définitions du concept de compétence dans le domaine des **sciences de l'éducation**. Cependant, il existe un certain consensus : une compétence est un savoir-faire qui intègre habileté et connaissance, qui est complexe, qui réfère à des habiletés cognitives, affectives, sociales ou psychomotrices et qui est spécifique à un ensemble de situations (donc à un contexte) (Lasnier, 2000).

Jonnaert (2002) adopte une approche en cascade et présente l'architecture d'une compétence comme suit :

Tableau I-1. Des mobilisations en cascade (d'après Jonnaert, 2002)

Au niveau de la compétence	Au niveau des capacités	Au niveau des habiletés	Au niveau des contenus disciplinaires
La <i>compétence</i> convoque une série de ressources pour traiter une situation avec succès. Certaines de ces ressources sont des capacités cognitives maîtrisées par le sujet.	Les <i>capacités</i> sélectionnées coordonnées entre elles reposent sur une série d'habiletés que le sujet maîtrise à propos de certains contenus disciplinaires.	Les <i>habiletés</i> sur lesquelles reposent les compétences et les capacités mobilisées, permettent une utilisation appropriée de certains contenus disciplinaires.	Les <i>contenus disciplinaires</i> fournissent la matière première aux habiletés et aux capacités.

La définition que le Ministère de l'Éducation du Québec a retenue en 2000 est la suivante : « Savoir-agir fondé sur la mobilisation et l'utilisation efficace d'un ensemble de ressources. » (Lasnier, 2000, p. 31). Cette définition ne suffit pas à expliquer et à comprendre le concept de compétence et surtout à l'opérationnaliser. Lasnier (2000, p. 32) propose donc une définition plus opérationnelle de compétence:

« Une compétence est un savoir-agir complexe résultant de l'intégration, de la mobilisation et de l'agencement d'un ensemble de capacités et d'habiletés (pouvant être d'ordre cognitif, affectif, psychomoteur ou social) et de connaissances (connaissances déclaratives) utilisées efficacement, dans des situations ayant un caractère commun. »

Le mot intégration est très important parce qu'il met en évidence le besoin de créer des liens entre les différents éléments d'une compétence. Voici les définitions de capacité et d'habileté (Lasnier, 2000, p. 33) :

« Une capacité est un savoir-faire moyennement complexe, intégrant des habiletés (cognitives, affectives, psychomotrices et sociales) et des connaissances déclaratives. Une habileté est un savoir-faire simple, intégrant des connaissances déclaratives. »

Paquette (2002b, p.189) propose une autre définition de compétence qui considère les capacités et les habiletés comme équivalentes :

« Une compétence est un énoncé de principe qui régit une relation entre un public cible (ou acteur), une habileté et une connaissance. Un profil de compétences est un ensemble de compétences concernant un même public cible. ».

Cette définition est illustrée à la figure I-1.

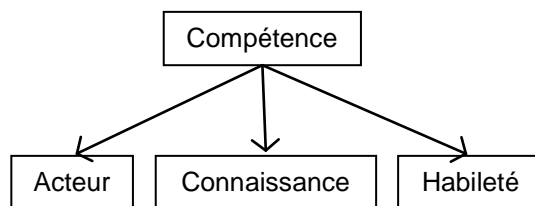


Figure I-1 Le concept de compétence

Les connaissances peuvent être des concepts, des procédures, des principes ou des faits. Les habiletés décrivent les processus qui peuvent être appliqués aux connaissances d'un domaine d'application pour les percevoir, les mémoriser, les assimiler, les analyser, les synthétiser, les évaluer, etc. (Paquette, 2002a). Les habiletés se situent donc au niveau métacognitif car ce sont des connaissances

agissant sur d'autres connaissances. Nous avons identifié deux types de compétences : les compétences de domaine (ou disciplinaires) et les compétences transversales. Les compétences disciplinaires sont liées à un domaine de connaissances particulier (par exemple les mathématiques) et les compétences transversales sont valables pour un ensemble de domaines différents (par exemple : résolution de problèmes, communication orale...). Un exemple de compétence disciplinaire est : « Utiliser le théorème de Pythagore pour résoudre des problèmes simples »; et un exemple de compétence transversale est : « Faire une présentation orale de ses idées devant un auditoire hétérogène ».

IMS Global Learning Consortium (2002) propose aussi une définition du concept de compétence. Le terme compétence est utilisé dans un sens large, incluant habileté, connaissance, tâche et produit de l'apprentissage. La définition contient quatre éléments :

- un identifiant (étiquette unique qui identifie la compétence)
- un titre (texte)
- une description (optionnel, texte interprétable uniquement par des humains)
- une définition.

Le dernier élément, définition, est le plus intéressant. Il peut être composé d'éléments qui peuvent être vus comme des attributs. La définition de compétence de Paquette peut facilement être exprimée sous la forme d'IMS. Pour cela, habileté et connaissance de Paquette seront des éléments de la définition d'une compétence selon IMS.

S'ajoutent à ces définitions de compétences des taxonomies. En effet, pour pouvoir classer les habiletés ou capacités, il faut avoir une taxonomie. Une taxonomie de capacités ou d'habiletés permet de classer des énoncés en fonction de leur degré de complexité. L'idée d'utiliser une taxonomie en éducation est reliée aux travaux de Bloom (1969) qui a créé des taxonomies d'objectifs éducatifs pour les différents domaines de l'apprentissage (cognitif, affectif, métacognitif et psychomoteur). Dans

le cadre de l'approche par compétences, ce sont les habiletés qui sont classées par des taxonomies. Beaucoup de taxonomies ont été développées dans le cadre de l'approche d'apprentissage par objectifs. Ces taxonomies ne sont pas complètement valables pour l'approche par compétences mais elles constituent une très bonne source d'inspiration (Lasnier, 2000). Les taxonomies de Bloom (1969) pour le domaine cognitif et celle de Krathwohl, Bloom et Masia (1969) pour le domaine affectif sont les plus utilisées car les plus connues. Ce sont d'ailleurs les taxonomies que Lasnier (2000) préconise pour l'application de l'approche par compétences du Ministère de l'Éducation du Québec.

Paquette (2002a) stipule que les habiletés peuvent être appréhendées de différentes manières, comme processus génériques de résolution de problèmes (classification en dix catégories), en tant que métaconnaissances actives (il existe beaucoup de classifications pour cette approche des habiletés) et la dernière approche est celle des domaines cognitif, affectif, social et psychomoteur. Paquette propose d'intégrer l'ensemble des taxonomies issues des travaux en sciences cognitives, en génie cognitif et en design pédagogique. Cette taxonomie comporte trois couches allant du général vers le particulier. La couche la plus générale correspond à quatre phases du cycle de traitement de l'information (recevoir, reproduire, produire ou créer et autogérer). La couche intermédiaire comporte 10 habiletés : prêter attention, intégrer, instancier ou préciser, transposer ou traduire, appliquer, analyser, réparer, synthétiser, évaluer et auto-contrôler.

Il est intéressant de noter que l'approche par compétence, qui offre un référencement sémantique riche, peut s'avérer utile pour adapter l'apprentissage. Plus précisément, ce référencement sémantique est indispensable à une adaptation des apprentissages pertinente au regard des attentes et besoins de l'apprenant. En effet, comme le stipule Paquette (2002a, p. 281) : « Dire qu'une personne doit connaître quelque chose ou acquérir telle ou telle connaissance n'est pas suffisant. Il faut se demander jusqu'à quel point, de quelle façon, dans quel contexte, pour quel usage. » Les compétences permettent de répondre à ces questions en associant une habileté à une connaissance. En effet, celles-ci précisent quel processus (cognitif, affectif, social

ou psychomoteur) fera le traitement de la connaissance. Ainsi, la compétence fournit non seulement un objectif d'apprentissage, mais aussi des moyens à mettre en œuvre pour acquérir les connaissances et les habiletés.

1.2.2. Les compétences selon Paquette

Nous allons travailler avec la définition proposée par Paquette (2002a). Comme nous venons de le voir, il existe d'autres propositions de représentation des compétences. Nous avons choisi celle de Paquette car elle nous semble mieux adaptée à notre recherche. En effet, elle est issue de l'étude d'un ensemble de travaux réalisés en éducation et elle est orientée ingénierie, c'est-à-dire qu'elle a été élaborée en vue de créer des systèmes d'apprentissage performants. Elle est aussi exprimable selon le standard RDCEO d'*IMS Global Learning Consortium* (Global Learning Consortium, 2002). Paquette (2007, p. 10) présente une ontologie (figure I-2) qui fournit une description du concept de compétence défini comme « la relation entre une habileté générique appliquée à une connaissance à un certain niveau de performance ». La légende du formalisme utilisé dans les figures suivantes est présentée à l'appendice A page 168.

Figure I-2. Ontologie du concept de compétence (tiré de Paquette, 2007)

Une compétence dispose d'un contexte d'utilisation (compétence pré-requise, compétence actuelle ou compétence cible), elle a un énoncé en langage naturel et elle peut permettre d'annoter une ressource (ressource dans son sens large, incluant les matériels pédagogiques mais aussi les acteurs). La compétence est aussi liée à une connaissance tirée d'une ontologie de domaine ainsi qu'à une habileté générique qui elle-même est liée à des indicateurs de performance. Pour plus de détails sur la signification de cette ontologie de compétence, se reporter à Paquette (2007).

Paquette (2007) propose aussi une extension de l'ontologie du concept de compétence aux habiletés génériques et une extension aux indicateurs de performance. La figure I-3 présente l'extension de l'ontologie du concept de compétence aux habiletés génériques.

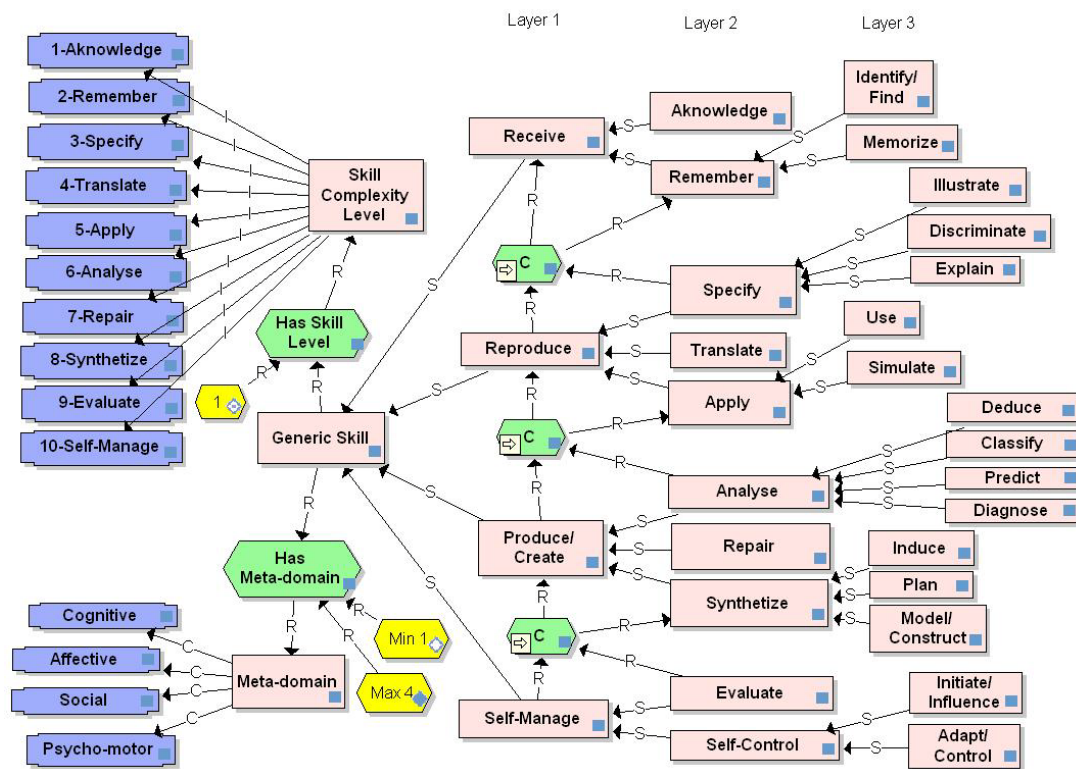


Figure I-3. Extension de l'ontologie de compétence aux habiletés génériques (tiré de Paquette, 2007)

Cette ontologie d'habiletés comporte trois couches de plus en plus spécifiques d'habiletés génériques qui peuvent être appliquées à différents domaines de connaissance. Une habileté générique possède un degré de complexité. La taxonomie proposée permet donc de classer les habiletés selon leur complexité cognitive. Cette taxonomie est un exemple, une autre taxonomie pourrait être utilisée avec l'ontologie de compétence présentée précédemment. Nous utiliserons cette taxonomie car sa création est née de l'intégration des travaux, entre autres, de Bloom, Romiszowski, Krathwohl, Merrill qui ont créé des taxonomies pour différents domaines d'apprentissage.

La figure I-4 présente l'extension de l'ontologie de compétence aux indicateurs de performance. Les indicateurs de performance sont associés à une habileté générique. Paquette (2007) propose cinq indicateurs de performance qui permettent de préciser l'habileté. Ceux-ci sont définis selon que l'habileté s'exerce sur la

connaissance : (1) de façon épisodique ou persistante (la fréquence), (2) de façon partielle ou globale (l'étendue), (3) de façon dirigée ou autonome (l'autonomie), (4) dans des situations simples, moyennes ou complexes (la complexité) et (5) dans des situations familières ou nouvelles (le contexte). Dans la pratique, ce référentiel d'indicateurs de performance pourrait être remplacé par un autre, comme le référentiel d'habiletés présenté précédemment. Nous avons fondé notre réflexion sur ces indicateurs de performance pour créer notre modèle de l'apprenant mais nous verrons au chapitre V qu'il est possible d'utiliser un autre référentiel pour évaluer le « niveau » ou plus exactement le degré de maîtrise d'une compétence.

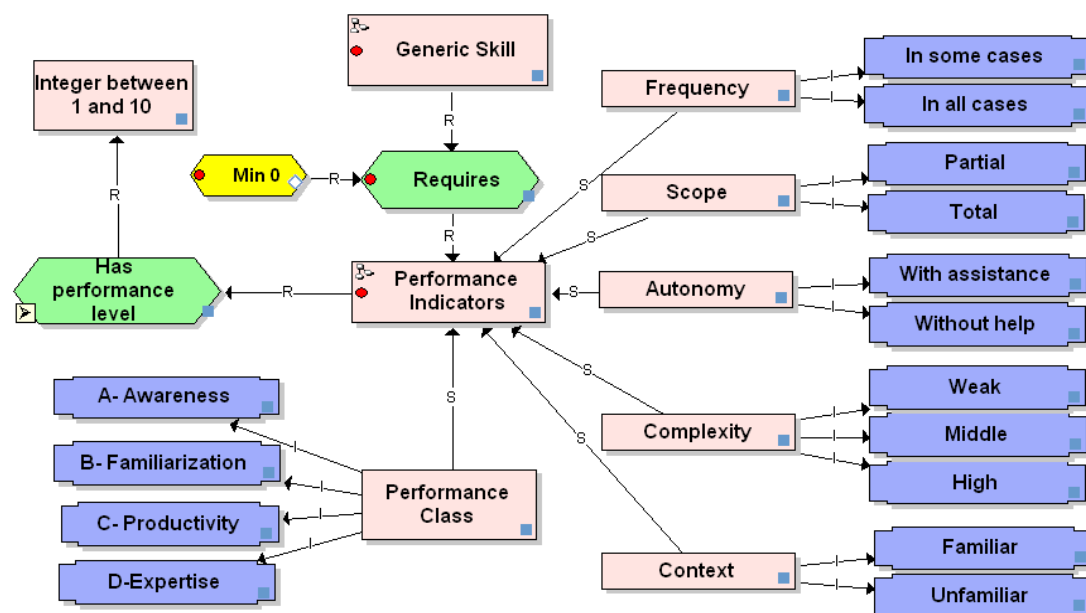


Figure I-4. Extension de l'ontologie de compétence aux indicateurs de performance (tiré de Paquette, 2007)

Le tableau I-2 présente les valeurs des indicateurs de performance associées à des niveaux de performance.

Tableau I-2. Progression dans la performance de l'exercice d'une compétence (d'après Paquette, 2002a)

Indicateur de performance	Niveaux de performance			
	Sensibilisation 0,0-2,5	Familiarisation 2,5-5,0	Maîtrise 5,0-7,5	Expertise 7,5-10,0
Fréquence	Épisodique	Persistante	Persistante	Persistante
Autonomie	Dirigé	Autonome	Autonome	Autonome
Étendue	Partielle	Partielle	Globale	Globale
Complexité	Simple	Simple	Moyen	Complexe
Contexte	Familière	Familière	Familière	Nouvelle

Paquette propose ainsi quatre niveaux de performance : *sensibilisation*, *familiarisation*, *maîtrise* et *expertise*. Dans le tableau, les éléments en gras sont les différences avec le niveau précédent. Par exemple, la différence entre les niveaux *sensibilisation* et *familiarisation* est la valeur des critères *fréquence* et *autonomie* qui passent des valeurs *épisodique* et *dirigée* à ***persistante*** et ***autonome***.

Dans la suite de cette recherche, nous avons adopté la vision de Paquette sur le concept de compétence ainsi que son référentiel d'habiletés génériques et ses indicateurs de performances.

1.3. Les différents acteurs intervenant au cours du processus d'apprentissage (La gestion de points de vue multiples)

Un des objectifs de cette recherche est de proposer un modèle de l'apprenant ouvert aux différents acteurs participant au processus d'apprentissage. Ce modèle doit permettre à ces acteurs d'intégrer leur vision, leur point de vue sur l'apprenant (ou plus précisément sur son état cognitif puisque notre modèle est un modèle cognitif). Nous avons ainsi décidé d'explorer les systèmes permettant la gestion de points de vue multiples car ce pourrait être une solution à intégrer dans notre modèle de l'apprenant.

La gestion d'un modèle prenant en compte différents acteurs qui ont chacun des perspectives ou vues différentes sur le modèle en question est un problème complexe puisque chaque acteur a une vue différente du modèle. De plus, les vues de chacun

des acteurs sont partielles ou incomplètes. La combinaison d'un acteur et de sa vue est appelée point de vue (Finkelstein et Sommerville, 1996). Un point de vue est la « manière qu'a quelqu'un d'envisager, de voir, de juger » (larousse.fr). Comme décrit dans Marino, Rechenmann et Uvietta (1990), un point de vue établit une vue partielle de l'objet observé; selon l'observateur, une partie des attributs observés, mais certainement pas tous, sont concernés par un point de vue. Pour Ribière (1999, p. 82), « un point de vue est l'interface permettant l'indexation et l'interprétation d'une vue composée d'éléments de connaissances, il est caractérisé par un point de focalisation et un angle de vue ». Dieng-Kuntz, Corby, Grandon, Giboin, Golebiowska, Matta et Ribière (2000) précisent qu'en termes de construction, le point de vue permet d'indexer des connaissances afin de les rendre accessibles, dynamiques et réutilisables et qu'en termes de consultation, il constitue un filtre permettant de n'afficher à l'utilisateur que les informations pertinentes à ses yeux. Ils expliquent aussi ce que sont le point de focalisation et l'angle de vue de la définition de Ribière. Le point de focalisation décrit le contexte et l'objectif du point de vue alors que l'angle de vue décrit les caractéristiques de ceux qui s'expriment selon un certain point de focalisation. Dieng-Kuntz *et al.* (2000) stipulent qu'on retrouve deux types de points de vue :

- (1) les points de vue définissant des vues « perspectives ». Ils indexent des descriptions consensuelles d'un même objet par différents acteurs, les vues sont alors complémentaires et forment une vision cohérente du monde.
- (2) les points de vue définissant des vues « opinions ». Ils indexent des vues non consensuelles relatant chacune une approche particulière d'un acteur, les vues représentent alors indépendamment les unes des autres des visions incomplètes du monde et peuvent collectivement être inconsistantes.

Dans notre projet, nous nous intéressons aux points de vue définissant des vues « **opinions** ». En effet, nous souhaitons que notre modèle permettent à chaque acteur impliqué d'exprimer ses opinions, de faire connaître son opinion sur les apprentissages de l'apprenant représenté par le modèle.

Selon Finkelstein et Sommerville (1996), le principal problème engendré par la prise en compte de points de vue est l'inconsistance ou l'incohérence. En effet, comme noté par Dieng-Kuntz *et al.* (2000), différents points de vue sur un même objet peuvent être consensuels, conflictuels, correspondants ou contrastants. Un point de vue consensuel apparaît lorsque deux ou plus observateurs regardant les mêmes attributs d'un objet y voient les mêmes valeurs. Au contraire, des points de vue conflictuels apparaissent lorsque deux observateurs regardant un même objet au même moment voient des valeurs différentes pour certains attributs. Les notions de correspondance et de contraste font référence à des différences sur la notation sémantique des concepts. Dans le cas de la correspondance, des acteurs différents utilisent des termes différents pour préciser un même objet ou caractéristique d'un objet et dans le cas du contraste, des acteurs utilisent des termes semblables pour désigner des objets différents. En ce qui concerne l'évaluation de compétences, des points de vue conflictuels vont très certainement apparaître. Nous aurons donc à gérer ces conflits dans notre modèle de l'apprenant si nous acceptons l'idée de créer des points de vue.

Benchikha, Boufaïda et Seinturier (2005) précisent que les points de vue sont de plus en plus utilisés ces dernières années dans le développement de systèmes informatiques. Les objectifs visés par l'utilisation de points de vue sont multiples. Ces objectifs sont :

- (1) point de vue comme moyen de fournir plusieurs descriptions d'une même entité;
- (2) point de vue comme moyen pour représenter et prendre en compte la complexité d'un système;
- (3) point de vue comme approche pour la modélisation et le développement distribué de systèmes;
- (4) point de vue comme mécanisme avancé pour les technologies orientées objet;
- (5) point de vue comme mécanisme pour résoudre des problèmes.

Aussi, lors de l'utilisation de points de vue dans un outil informatique, trois problèmes apparaissent : (1) choisir un bon modèle de point de vue, (2) identifier les

différents points de vue et (3) gérer les informations issues des différents points de vue (Finkelstein et Sommerville, 1996).

Ces objectifs et problèmes listés ci-dessus sont liés au développement de systèmes informatiques et pour répondre à nos objectifs de recherche, nous souhaiterions plutôt utiliser la notion de point de vue au cours de l'utilisation d'un système informatique. Malheureusement, il semble n'exister que très peu de systèmes offrant la possibilité d'avoir différents point de vue au cours de l'utilisation, en tout cas, nous n'avons pas trouvé d'articles scientifiques traitant des difficultés liées à ce type d'utilisation de point de vue. Tout de même, ces objectifs et problèmes vont guider notre travail d'intégration de points de vue dans notre modèle de l'apprenant. En effet, les points de vue semblent être une excellente solution pour permettre une ouverture de notre modèle de l'apprenant aux différents acteurs participant au processus d'apprentissage, ces points de vue étant des vues partielles et incomplètes d'un même objet (l'apprenant dans notre cas). Nous verrons dans le chapitre III comment cette intégration est réalisée.

1.4. Conclusion

Dans ce chapitre, nous avons abordé différentes notions constituant notre cadre conceptuel. Tout d'abord, nous avons exploré les concepts de modèle et de métamodèle; de modèle puisque ce concept concerne directement notre problématique consistant à créer un modèle de l'apprenant et de métamodèle car cette notion nous permet de clarifier l'exposé de nos idées. Ce que nous pouvons retenir est qu'un modèle présente une vue partielle d'un objet ou d'un système, c'est-à-dire ne présentant que certaines des caractéristiques de l'objet ou système représenté. C'est le métamodèle qui précise quelles sont ces caractéristiques. On peut ainsi dire que le métamodèle est un filtre appliqué sur le monde réel et que le modèle est le résultat de l'application de ce filtre.

Nous avons ensuite exploré le concept de compétence car nous proposons d'adopter une approche par compétences. En effet, les compétences offrent un référencement sémantique riche parce qu'elles permettent de connaître ce qu'une personne sait (avec la connaissance) mais aussi à quel degré elle maîtrise cette connaissance dans un usage en contexte (grâce à l'habileté). Ceci fournit des moyens à mettre en œuvre pour la construction des connaissances par l'apprenant. Le référencement sémantique des acteurs que nous proposons d'adopter va nous permettre de créer un modèle de l'apprenant qui sera sémantiquement riche, formel (interprétable par des humains mais aussi par des machines) et interopérable (utilisable avec différents systèmes d'apprentissage en ligne).

Finalement, nous avons discuté de la prise en compte de différents acteurs. Nous avons plus particulièrement étudié la notion de point de vue qui semble prometteuse pour répondre à notre besoin d'ouverture du modèle de l'apprenant aux différents acteurs participant au processus d'apprentissage. Un point de vue est une vue partielle et incomplète d'un objet (dans notre cas, l'apprenant) par un acteur ou un groupe d'acteur. Nous souhaitons utiliser ces points de vue au cours de l'utilisation d'un système informatique et non lors de sa conception comme cela est décrit dans la plupart des articles existants sur le sujet. Nous souhaitons créer un modèle de l'apprenant permettant d'intégrer les visions que chacun des acteurs participant au processus d'apprentissage pourront avoir sur l'apprenant représenté par le modèle en question. L'intégration de points de vue à notre modèle est la solution que nous envisageons.

Nous verrons dans le chapitre III comment les notions présentées ici sont utilisées ou même intégrées à notre proposition de modèle de l'apprenant. Mais d'abord, nous explorons dans le chapitre II un état de la question des concepts et approches technologiques que nous utilisons dans cette recherche.

Chapitre II - CONTEXTE TECHNOLOGIQUE : ETAT DE LA QUESTION

Nous présentons dans ce deuxième chapitre l'état de la question des concepts et approches technologiques en lien avec notre recherche. Ce chapitre, tout comme le chapitre précédent, présente les résultats de l'étape 2 de la phase 1 de notre démarche de recherche, soit l'analyse de la situation. Tout d'abord, nous explorons l'ePortfolio, aussi bien d'un point de vue conceptuel que d'un point de vue outil (section 2.1.) car c'est ce qui nous permettra de faire le lien entre les compétences (présentées au chapitre I) et les productions de l'apprenant. Puis, nous étudions le concept de contrat (section 2.2.), notamment concernant les interactions du modèle de l'apprenant avec d'autres systèmes informatiques, pour répondre à notre objectif de modèle utilisable dans différents contextes et tout au long de la vie. Nous étudions ce concept de contrat d'un point de vue conceptuel, en nous inspirant des travaux informatiques effectués sur cette notion. Finalement, nous explorons une problématique particulièrement importante liée au volet informatique de ce projet de thèse : la gestion de versions (section 2.3.) qui permet elle aussi de répondre à l'objectif d'ouverture de notre modèle de l'apprenant à divers contextes et à celui d'apprentissage tout au long de la vie.

2.1. *ePortfolio*

Une nouvelle façon de collecter, d'organiser et de partager les productions des apprenants est l'utilisation d'un *ePortfolio*. Nous avons analysé dix définitions d'*ePortfolio* trouvées dans la documentation scientifique (Moulet, 2006a) selon deux

critères : le contenu et l'utilisation de l'*ePortfolio*. Notons que les écrits scientifiques plus récents confirment et appuient cette analyse datée de 2006 (Buzzeto-More, 2010, Cambridge, 2010, Cambridge, Cambridge et Yancey, 2009, Cheal, 2010 et Zubizarreta, 2009). Le contenu se divise en deux sous-éléments : la forme et le fond ou encore la structure et la sémantique. Le contenu peut être un ensemble d'informations numériques, ce qui en représente la structure, la forme. Ces informations peuvent représenter les résultats d'apprentissage d'une personne, ce qui est la sémantique, le fond du contenu. L'utilisation se divise aussi en deux sous-éléments : les services offerts et les buts d'un *ePortfolio*. Par exemple, un service peut être de permettre la création de présentations de son *ePortfolio* pour différentes audiences alors qu'un but peut être de faire réfléchir l'apprenant sur ses apprentissages. Les buts ou utilités choisis pour un *ePortfolio* vont avoir une incidence sur le contenu et sur les services proposés. L'analyse des définitions étudiées selon les critères Contenu et Usage se retrouve en détail à l'appendice B page 172.

Nous pouvons conclure de cette analyse qu'il n'existe pas de consensus concernant le contenu ou l'utilisation d'un *ePortfolio*. Toutefois, toutes les définitions s'accordent pour voir l'*ePortfolio* comme un profil personnel et une collection (numérique) des productions de l'apprenant. Par contre, le sous-critère sémantique ne fait pas l'objet d'un consensus, en effet, le contenu d'un *ePortfolio* peut porter sur ses apprentissages, ses formations, ses buts, ses expériences professionnelles, ses réussites, ses expériences, etc. Concernant le critère contenu, nous adoptons la définition d'IMS Global Learning Consortium (2004) qui stipule que les *ePortfolios* sont : « *collections of personal information about a learner that represent accomplishments, goals, experiences, and other personalized records that a learner can present to schools, employers, or other entities* ». Concernant le deuxième critère (utilisation), quatre usages majeurs se démarquent dans les définitions étudiées : (1) décrire et démontrer, (2) planifier, (3) réfléchir sur et (4) partager.

Nous avons identifié trois types d'*ePortfolio* dans la documentation (*IMS Global Learning Consortium*, 2004 et *ePortConsortium*, 2003). L'**ePortfolio personnel** est utilisé pour l'autoréflexion, il peut servir de journal de bord (concernant les expériences), il permet d'organiser des matériaux présentés en classe et il aide les étudiants à reconnaître leurs habiletés et à prendre des décisions. L'**ePortfolio d'apprentissage** est utilisé pour montrer l'apprentissage des étudiants, il fournit un cadre pour évaluer les progrès d'apprentissage et il permet de démontrer comment les habiletés se développent dans le temps. Pour finir, l'**ePortfolio professionnel** aide à faire des décisions de carrière, il peut démontrer que quelqu'un rencontre les exigences d'un programme, il peut aussi être utilisé pour présenter les habiletés et les accomplissements pour l'employabilité ou alors pour revoir le développement professionnel pour les avancées de carrière.

Nous pouvons constater que pour ces trois types d'*ePortfolio*, les quatre utilisations majeures identifiées précédemment sont valables (décrire et démontrer, planifier, réfléchir sur et partager). Le contenu et les services dépendent en partie du type d'*ePortfolio* développé mais principalement des buts recherchés pour l'outil. Avant de commencer le développement d'un outil *ePortfolio*, il faut donc se poser les questions suivantes : Quel type d'*ePortfolio* ? Pour qui ? POURQUOI ? Quel rôle pouvons-nous envisager pour un tel outil dans notre modèle de l'apprenant ? Un outil *ePortfolio* offre des fonctionnalités très intéressantes et nous semble pertinent pour notre modèle de l'apprenant. Un tel outil nous permettra de disposer des produits de l'apprenant (créés au cours des apprentissages) et ainsi de créer un lien entre ces produits et les compétences développées au cours du temps par l'apprenant. Plus précisément, les caractéristiques des outils *ePortfolio* qui nous intéressent sont les suivantes :

- l'*ePortfolio* est un outil développé pour l'apprenant lui-même et est visible et modifiable par des acteurs humains;
- l'*ePortfolio* contient les productions de l'apprenant;

- les outils *ePortfolios* offrent la possibilité d'observer l'évolution des apprentissages.

C'est avec ces notions en tête que nous avons étudié des outils *ePortfolio* existants. Pour faciliter la lecture, nous exposons cette étude ici bien qu'elle corresponde à l'étape 2.3 (Implémentation et test) de la phase 2 (Élaboration et construction) de la démarche de recherche (et non à la phase 1 comme le reste des résultats présentés dans ce chapitre). En effet, cette étude a été menée afin de savoir si nous pourrions utiliser un outil existant pour implémenter notre prototype. Notre étude a aussi été dirigée par les caractéristiques du modèle de l'apprenant que nous souhaitons proposer, soit un modèle cognitif de l'apprenant, intégrant des compétences et des productions de celui-ci, un modèle évolutif (représentant l'évolution des apprentissages dans le temps) et un modèle multi-acteurs (ouverts aux différents acteurs prenant part au processus d'apprentissage). En effet, nous avons analysé les outils *ePortfolios* en cherchant à nous assurer que ces caractéristiques de notre modèle de l'apprenant étaient présentes dans l'outil ou pourraient être ajoutées par la suite à l'outil étudié.

Nous avons étudié plus particulièrement l'outil *Open Source Portfolio* (OSP) et ce pour deux raisons : d'abord, c'est un *ePortfolio* qui semblait, à première vue, pouvoir répondre à nos besoins de conception d'un modèle de l'apprenant et ensuite parce qu'il est distribué sous licence libre (*open source*). Nous avons trouvé d'autres outils *ePortfolio*, mais la plupart ne sont pas sous licence libre (*ePortfolio* du Connecticut Distance Learning Consortium (CTDLC), Edu-portfolio de Université de Montréal sous la supervision du professeur Thierry Karsenti, EUCEBS (*EUropean CErtificate Basic Skills*) Edinburgh University Settlement...). Finalement OSP était le seul outil sous licence libre qui semblait correspondre à notre projet, soit la création d'un modèle de l'apprenant intégrant des compétences et des productions de l'apprenant, pouvant les lier entre elles, étant évolutif et à points de vue multiples. Notons que depuis cette étude de nombreux *ePortfolio* sous licence libre ont vu le jour. D'ailleurs, Cheal (2010) en analyse quelques-uns (dont OSP), comme Mahara et

MyStuff et Exabis qui sont des modules de Moodle (un système d'apprentissage en ligne). Il en existe d'autre comme Cyberfolio et Elgg par exemple.

2.1.1. L'outil OSP

L'analyse complète de cet outil se trouve à l'appendice C page 175. Dans cette section, nous allons discuter des avantages et inconvénients d'OSP. Le principal avantage d'OSP est sa richesse. Cet outil comporte en effet de nombreuses fonctionnalités et permet à un utilisateur (administrateur) de choisir les fonctionnalités qu'il souhaite offrir à son public cible. De plus, OSP est un logiciel sous licence libre, chacun peut donc le télécharger et l'utiliser gratuitement. OSP est développé dans un environnement universitaire de recherche, il est donc facilement possible de s'intégrer à la communauté OSP pour discuter d'idées de nouvelles fonctionnalités et éventuellement les développer. Malgré ces avantages, OSP ne nous paraît pas être utilisable pour notre projet de modélisation de l'apprenant et ce pour trois raisons différentes.

Premièrement, cet outil ne prend pas en compte le concept de compétence, et aucune porte ne semble ouverte pour l'intégrer. Cela n'est pas obligatoirement un problème, en effet, OSP étant sous licence libre, chacun peut le modifier selon ses besoins. Comme les compétences sont un aspect important de notre modèle, nous ne voulons pas uniquement ajouter un module ou une fonctionnalité "compétence" : notre modèle est **fondé** sur le concept de compétence (entre autre).

Deuxièmement, même si OSP est modifiable par son essence même de gratuité, cela semble difficile à faire. En effet, il n'existe pas ou presque pas de documentation fonctionnelle ou technique sur OSP. Ainsi, il est difficile d'une part d'installer et d'utiliser l'outil mais aussi de comprendre son fonctionnement technique afin d'être capable de le modifier.

Troisièmement, l'utilisabilité¹ n'est pas à la hauteur de ce que nous souhaitons avoir pour notre outil. Nous avons principalement étudié l'utilisabilité des fonctionnalités liées à la création de présentation. Il est difficile de créer une présentation : il faut comprendre la structure d'OSP pour pouvoir le faire et passer par différentes étapes. Il faut d'abord créer des instances de formulaires et seulement ensuite créer une présentation. Pour un nouvel utilisateur, le temps d'apprentissage sera donc long, il devra suivre un didacticiel pour pouvoir créer des présentations et ceci ne convient pas à notre projet. Malgré certains avantages de cet outil, nous ne l'utilisons pas à cause des inconvénients décrits précédemment.

2.2. Le concept de contrat

La notion de contrat est utilisée en informatique en référence aux interactions qu'il peut y avoir entre différents composants informatiques. Elle nous intéresse donc dans le cadre de notre projet de thèse car nous souhaitons disposer d'un modèle de l'apprenant ouvert et interopérable qui pourra être en relation avec différents systèmes d'apprentissage en ligne. Selon le Petit Robert (version électronique 2.1), un contrat est une « convention par laquelle une ou plusieurs personnes s'obligent, envers une ou plusieurs autres, à donner, à faire ou à ne pas faire quelque chose ». Plus spécifiquement dans le domaine informatique, un contrat est « l'ensemble des éléments liant un demandeur et un fournisseur et définissant d'une manière précise, complète et cohérente, leurs obligations respectives » (Grand dictionnaire terminologique de l'Office québécois de la langue française). La notion de contrat fait ainsi référence à trois autres principales notions : le demandeur, le fournisseur et leurs obligations.

¹ L'utilisabilité est, selon la norme ISO 9241, "le degré selon lequel un produit peut être utilisé par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d'utilisation spécifié".

En informatique, les contrats déterminent les relations entre différents acteurs (ou composants informatiques ou systèmes ou modèles, etc.) (Bachman, Bass, Buhman, Comella-Dorda, Long, Robert, Seacord et Wallnau, 2000).

Bachman *et al.* définissent la notion de contrat ainsi :

- un contrat est établi entre deux parties ou plus;
- les parties négocient souvent les détails du contrat avant sa signature;
- les contrats prescrivent des comportements normatifs et mesurables concernant tous les signataires;
- les contrats ne peuvent être changés à moins que les changements soient approuvés par tous les signataires.

Les signataires du contrat sont co-dépendants et ont des obligations réciproques. La notion de contrat met en évidence non pas les intervenants (le demandeur et le fournisseur) mais plutôt leurs interactions.

Différents types de contrats sont répertoriés dans la documentation scientifique. Selon Beugnard, Jézéquel, Plouzeau et Watkins (1999), les contrats peuvent être divisés en quatre niveaux de propriétés de plus en plus négociables. Le premier niveau est le niveau des contrats basiques ou syntaxiques. Il est requis pour que le système fonctionne. Le second niveau, les contrats comportementaux, améliore le degré de confiance dans un contexte séquentiel. Il spécifie les comportements du fournisseur et les informations que doit fournir le client ou demandeur. Le troisième niveau, celui des contrats de synchronisation, améliore le degré de confiance dans des contextes distribués ou concurrentiels. Ce niveau permet de décrire les dépendances entre les services offerts par le fournisseur. Il garantit que, quel que soit le client qui demande un service au fournisseur, ce service est correctement exécuté. Le dernier niveau, les contrats de qualité de service, permet de quantifier la qualité de service et est généralement négociable. Pour Villalobos (2003), un contrat a plutôt trois niveaux : un niveau lexical, un niveau syntaxique et un niveau sémantique. Le niveau lexical (ou fonctionnel) permet de vérifier la compatibilité entre la requête du client et le fournisseur. Un client ne peut pas se connecter à un fournisseur s'il ne fournit pas une

requête « compatible » avec le langage du fournisseur. Le niveau syntaxique (ou contrat d'interaction ou comportemental) décrit le protocole d'utilisation des services offerts par le fournisseur. Ce contrat permet de respecter la forme d'utilisation des services. Le dernier niveau est le niveau sémantique, il décrit les conséquences pour un client de contacter un fournisseur pour un service particulier.

La notion de contrat s'est donc révélée être effectivement très pertinente pour notre projet pour la gestion des interactions de notre modèle de l'apprenant avec d'autres systèmes informatiques. En effet, comme nous adoptons une vision d'apprentissage tout au long de la vie, notre modèle de l'apprenant sera très probablement amené à se brancher à différents systèmes. Nous avons travaillé au niveau conceptuel et sémantique des contrats. Nous ne présenterons donc pas plus en détail les applications techniques de ce concept. Cette notion nous a été indispensable pour concevoir l'interaction de notre modèle de l'apprenant avec des systèmes d'apprentissage en ligne, nous préciserons donc plus en détail dans le chapitre III comment nous concevons et utilisons cette notion de contrat dans notre recherche.

2.3. La gestion de versions

Notre modèle de l'apprenant est évolutif au même titre que l'apprentissage. Plusieurs versions sont donc créées au fur et à mesure que l'apprentissage a lieu. Ces versions doivent être gérées et c'est pour cela que nous nous intéressons à la gestion de versions. La gestion de versions en informatique est un problème difficile. Les logiciels de gestion de versions ont surtout été utilisés par les programmeurs informatiques. Mais de plus en plus d'acteurs les utilisent maintenant dans d'autres domaines, par exemple pour l'écriture collaborative de texte. Les logiciels de gestion de versions sont un outil très important du travail collaboratif (Bortzmeyer, 2005). Ils permettent d'ajouter une dimension temporelle au système de gestion de fichiers traditionnel. En effet, ces logiciels gardent trace des différentes versions d'un fichier,

ce qui permet d'en connaître l'historique et d'en récupérer n'importe quelle version. Ainsi, il est possible de revenir en arrière en cas d'erreur, d'identifier les modifications apportées entre deux versions, de travailler en équipe sur un projet, de gérer des branches de développement en parallèle, etc. Selon Kilpi (1997, p. 307), un outil de gestion de versions :

« manages and keeps track of the configuration items which are any documents created during a software development process, and which are found necessary to be placed under configuration control like requirements documents, data flow diagrams, design documents, source code, and test results ».

De plus, Conradi et Westfechtel (1998) spécifient qu'un modèle de version définit les objets qui seront « versionnés », l'identification et l'organisation des versions, ainsi que les opérations effectuées pour récupérer les versions existantes et pour construire de nouvelles versions.

Les différentes versions d'un fichier, d'un logiciel ou de tout autre produit informatique sont liées par des modifications. Une nouvelle version sera donc différente de l'ancienne. Les modifications peuvent être des ajouts, des suppressions, des modifications ou un ensemble des trois. Les logiciels de gestion de versions permettent de retrouver les modifications effectuées entre une version N et une version N+1. Un changement de version peut correspondre en fait à plusieurs modifications. D'ailleurs le terme de révision est parfois utilisé pour pallier la confusion entre une modification apportée dans un fichier et une modification dans le sens de la gestion de versions. Dans le cas d'un logiciel, une nouvelle version peut correspondre à des modifications apportées à plusieurs fichiers. Lorsque cette gestion de versions est utilisée pour le travail collaboratif, les membres de l'équipe peuvent travailler de façon indépendante et en parallèle. Des conflits de révisions peuvent donc apparaître. Pour chaque personne travaillant sur le projet, une branche est créée, il faut ensuite effectuer une fusion de ces branches pour obtenir une nouvelle version globale de l'outil informatique. Mais certaines révisions peuvent être contradictoires. Il y a alors conflit et le logiciel de gestion de versions n'est pas en mesure de savoir

quelle est la révision à conserver. Un mécanisme de consensus ou de gestion de conflit peut alors être utilisé entre les membres de l'équipe.

Les fonctionnalités de base des logiciels de gestion de versions sont le marquage (consiste à associer un nom à chaque version), la comparaison (des versions pour identifier les modifications apportées) et éventuellement le verrouillage et les notifications lorsque le travail est fait en équipe. Le verrouillage empêche les membres de l'équipe de modifier un fichier lorsque celui-ci est utilisé par un des membres et la notification émet un avis de révision aux autres membres de l'équipe. Les logiciels de gestion de versions stockent également des métadonnées sur les révisions (comme par exemple, l'auteur des modifications et leur date).

Il existe deux principaux types de système de gestion de versions : les systèmes centralisés et les systèmes décentralisés. En gestion centralisée, un serveur central sert de dépôt et le client se connecte à ce dépôt pour récupérer en local un fichier sur lequel il veut travailler. Ensuite, le client dépose le fichier modifié sur le serveur central. CVS, *Current Version System*, a été le système centralisé le plus utilisé pendant très longtemps et il fait encore référence aujourd'hui (Bortzmeyer, 2005, Shaikh et Cornford, 2003). Des conflits peuvent apparaître lorsque plusieurs personnes travaillent en même temps sur le même fichier. La solution simpliste à ce problème est le verrouillage : lorsqu'une personne travaille sur un fichier, personne d'autre ne peut y accéder. Mais dans le cadre des projets d'envergure ou pour le travail collaboratif, cette solution est trop contraignante. Une autre solution consiste donc à faire régulièrement des fusions entre les différentes versions d'un fichier. Dans ce cas, il est nécessaire de gérer les conflits. La gestion de versions centralisée comporte aussi certaines limites. Par exemple, il n'est pas possible de travailler sans être connecté au réseau (car il faut pouvoir accéder au serveur central), il y a aussi des problèmes liés aux droits d'accès au dépôt (souvent certains utilisateurs sont « plus importants » que d'autres, ils ont la priorité ou alors leurs versions sont prises par défaut lors de la gestion de conflit, la gestion de droits d'accès spécifiques n'est pas

triviale), ceci peut poser des problèmes dans le cadre de travail coopératif et en équipe.

La solution proposée pour dépasser ces limites est la gestion de versions décentralisée. Dans ce cas, il n'y a plus de dépôt central, celui-ci est remplacé par un réseau de dépôts (tous équivalents, c'est-à-dire tous de même importance). Cette approche se veut plus équitable, plus pair-à-pair que l'approche centralisée. Elle permet aussi de s'abstraire de la séparation physique entre les dépôts. De plus, elle facilite la création de différentes branches de développement ainsi que leur intégration (ou fusion). Il existe différents systèmes de gestion de versions décentralisés comme, notamment, GNU Arch, Bazaar, Darcs, Mercurial, Monotone.

Notre modèle de l'apprenant est évolutif au même titre que l'apprentissage. Plusieurs versions sont donc créées au fur et à mesure que l'apprentissage a lieu. Ces versions doivent être gérées et c'est pour cela que nous nous intéressons à la gestion de versions. Nous avons choisi pour notre recherche d'approfondir notre réflexion sur la gestion de versions et sur ce qu'elle pourrait apporter à notre modèle de l'apprenant. Nous avons donc étudié divers logiciels de gestion de versions dont nous présentons maintenant l'analyse.

2.3.1. Étude, analyse et comparaison de logiciels de gestion de versions

Comme pour l'étude des outils ePortfolio, nous plaçons cette section ici bien qu'elle corresponde à l'étape 2.3 (Implémentation et test) de la phase 2 de la démarche de recherche dont les résultats sont présentés dans le chapitre IV. La gestion de versions est un domaine de recherche très étendu. Il existe d'ailleurs de nombreux logiciels de gestion de versions. Beaucoup sont développés pour gérer du code informatique, mais ils peuvent aussi permettre de gérer des versions de fichiers. Dans notre projet, ce type d'outil nous semble très pertinent. En effet, l'intégration d'un outil de gestion de versions permettrait l'opérationnalisation de la caractéristique évolutive de notre modèle de l'apprenant. Comme il existe de nombreux outils, nous

souhaiterions en intégrer un existant plutôt que développer un nouveau système de gestion de versions. Nous avons effectué une étude et une analyse de logiciels de gestion de versions existants afin de déterminer s'il existe un outil que nous pourrions réutiliser et si oui quel outil existant correspond le mieux aux besoins de notre projet. Nous avons orienté notre étude sur la comparaison proposée par *Better SCM Initiative* (2007) que nous avons complétée avec des informations obtenues sur les sites web des outils analysés. Nous avons construit un tableau comparatif comportant 37 critères classés en quatre catégories : information générale, information technique, fonctionnalités et interfaces utilisateur, et ce pour une liste de 37 logiciels de gestion de versions. Ce tableau se trouve à l'appendice D page 182. Nous ne présenterons ici que les critères (17) sur lesquels s'est fondée notre analyse (la description de l'ensemble des critères se trouve à l'appendice D). Ces critères sélectionnés sont ceux qui nous semblaient pertinents pour effectuer un choix parmi les différents systèmes existants. Un seul critère était discriminant, celui de la licence. Ainsi, nous avons éliminé de notre analyse tous les systèmes qui n'étaient pas sous licence libre, 16 systèmes ont donc été retenus pour notre comparatif.

2.3.1.1. Comparaison de divers logiciels de gestion de versions

Nous allons présenter ici les 17 critères d'analyse retenus ainsi que le tableau comparatif des 16 logiciels de gestion de versions sous licence libre.

2.3.1.1.1. Critères d'analyse

Les tableaux qui suivent présentent les critères à la base de notre analyse.

Tableau II-1 Critères d'analyse des systèmes de gestion de versions – Catégorie Information générale

Critère	Description
Statut de développement	Décrit le statut courant de développement du logiciel.
Modèle du répertoire	Décrit la relation entre les différentes copies présentes dans le répertoire. Deux types de modèle : client-serveur (centralisé) et distribué (décentralisé).
Modèle de l'accès simultané	Décrit comment les changements sont gérés afin d'éviter les inconsistances. Deux types de modèle : <i>lock</i> et <i>merge</i> .
Licence	Spécifie la licence du logiciel.
Licence libre	Spécifie si le logiciel est sous licence libre ou non.
Portabilité	Décrit si le logiciel est utilisable sur différents systèmes d'exploitation, architectures et autres types de systèmes.
Plateformes supportées	Spécifie les systèmes d'exploitation supportés par le logiciel.

Tableau II-2 Critères d'analyse des systèmes de gestion de versions – Catégorie Information technique

Critère	Description
Langage de programmation	Spécifie le langage de programmation utilisé pour le développement de l'application.
Documentation disponible	Décrit la qualité de la documentation disponible sur le logiciel.
Facilité de déploiement	Décrit la facilité avec laquelle on peut déployer le logiciel.

Tableau II-3 Critères d'analyse des systèmes de gestion de versions – Catégorie Fonctionnalités

Critère	Description
Validation atomique	Garanti que tout changement fait sera fusionné, sinon annulé. Garanti que si un changement est interrompu, le répertoire ne sera pas inconsistant.
Définir des permissions	Spécifie s'il est possible de définir des permissions d'accès pour des parties du répertoire.
Ensemble de changements	Spécifie si le répertoire supporte les ensembles de changements (façon de grouper un ensemble de modifications liées en un élément atomique).
Historique du fichier ligne par ligne	Spécifie si le système de gestion de versions a une option permettant de suivre l'historique du fichier ligne par ligne.
Signature des révisions	Fait référence à l'intégration d'une signature digitale aux changements.
Tags	Précise s'il est possible de donner des noms aux changements.
Support international	Précise si le logiciel a un support multi-langues.

Tous ces critères n'ont pas la même importance selon nos besoins. Nous avons défini trois catégories d'importance des critères. La première catégorie regroupe les critères **déterminants**, soit les plus importants pour guider notre choix d'outil de gestion de versions. La deuxième catégorie d'importance regroupe les critères **importants** pour notre choix, sans que ceux-ci soient déterminants. Et finalement, la dernière catégorie correspond aux critères **intéressants** qui peuvent orienter notre choix si plusieurs outils répondent aux critères déterminants et importants. Il est à noter que ces critères sont sélectionnés selon leur importance sur n'importe quel aspect du projet, conceptuel, informatique ou même pratique.

Nous allons maintenant expliquer pourquoi ces différents critères nous importent et la catégorie à laquelle ils appartiennent. Comme dit précédemment, le critère *Licence* (et plus spécifiquement *Licence libre*) est un critère discriminant. En effet, nous sommes à la recherche d'un outil de gestion de versions gratuit et modifiable que nous pourrions réutiliser dans le cadre de notre projet.

Six critères appartiennent à la catégorie **critères déterminants**. Les voici présentés ci-dessous sans ordre d'importance entre eux.

(1) *Modèle de l'accès simultané* de la catégorie *Information générale*. Ce critère précise comment les changements sont gérés afin d'éviter des inconsistances. Il existe deux types de modèle : *lock* qui bloque l'accès au répertoire lorsqu'une personne l'utilise et *merge* qui permet à plusieurs personnes de travailler en même temps sur le répertoire, dans ce cas, des branches parallèles sont créées puis fusionnées. Nous avons obligatoirement besoin d'un outil travaillant selon le modèle *merge* car notre modèle de l'apprenant sera multi-utilisateurs et nous voulons que plusieurs personnes puissent accéder et modifier un même modèle en même temps.

(2) *Langage de programmation* de la catégorie *Information technique*. En effet, nous utilisons différents outils dans ce projet (comme nous le verrons dans les chapitres suivants) et ils doivent être compatibles entre eux. Cela nécessite qu'ils soient utilisables à partir d'un environnement en langage de programmation Java. De plus, nous voulons être capables d'adapter l'outil choisi aux besoins spécifiques de

notre projet. Pour cela, nous devons être en mesure de le modifier. L'outil doit donc être développé dans un langage de programmation que nous maîtrisons, soit idéalement le Java.

(3) *Portabilité* de la catégorie *Information générale*. Ce critère précise le degré de portabilité de l'outil (excellent, moyen, etc.), c'est-à-dire si l'outil est utilisable sur différents systèmes d'exploitation, architecture et autres types de systèmes. Ce critère ne précise pas quels sont les systèmes d'exploitation en question (critère 5). Comme nous souhaitons adapter et intégrer l'outil choisi à notre propre système, il est déterminant que celui-ci ait une grande portabilité. En effet, une grande portabilité signifie que nous aurons plus de chance de pouvoir facilement intégrer l'outil à notre système, sans avoir à faire de choix technique pour notre système en fonction des possibilités offertes par l'outil de gestion de versions sélectionné.

(4) *Définir des permissions* de la catégorie *Fonctionnalités*. Ce critère spécifie s'il est possible de définir des permissions d'accès pour des parties du répertoire. Ce critère nous intéresse parce que notre système est multi-utilisateurs et parce que tous les utilisateurs n'auront pas les mêmes droits d'accès aux informations contenues dans un modèle de l'apprenant. Il est donc important de pouvoir définir des permissions d'accès différentes pour les différents utilisateurs.

(5) *Plateformes supportées* de la catégorie *Information générale*. Ce critère complète le critère *Portabilité*. Il précise quels systèmes d'exploitation sont supportés par le logiciel. Nous avons besoin d'un outil qui supporte obligatoirement Windows car c'est sur ce système d'exploitation que nous allons effectuer le développement de notre système.

(6) *Validation atomique* de la catégorie *Fonctionnalités*. Ce critère garantit que tout changement sera fusionné totalement et si ce n'est pas possible, le changement ne sera pas du tout effectué. Cela est principalement important lorsqu'une modification est interrompue, puisque dans ce cas, ce critère garantit que le répertoire ne sera pas laissé dans un état inconsistant. Donc si les changements sont *atomic*, alors obligatoirement la totalité des changements (ou modifications) soumis par un

utilisateur est effectué. Cela prévient les inconsistances en cas d'interruption lors de la soumission du changement. Ceci est important pour nous puisque ce critère garantit en partie la consistance du répertoire.

Les **critères importants** (deuxième catégorie d'importance) sont au nombre de quatre.

(1) *Modèle du répertoire* de la catégorie *Information générale*. Ce critère décrit la relation entre les différentes copies présentes dans le répertoire. Il existe deux types de modèle : client-serveur (centralisé) et distribué (décentralisé). Pour le développement d'un système comme le notre, ce critère est a priori important. Bien que de moindre importance de par l'envergure de notre projet de recherche, ce critère demeure important dans l'optique d'une utilisation plus large.

(2) *Facilité de déploiement* de la catégorie *Information technique*. Comme pour le critère précédent, ce critère est important dans le cas d'applications qui seront déployées à grande échelle. En effet, si l'on souhaite déployer un outil, il importe que celui-ci soit facile à déployer.

(3) *Historique du fichier ligne par ligne* de la catégorie *Fonctionnalités*. Ce critère spécifie si le système de gestion de versions a une option permettant de suivre l'historique du fichier ligne par ligne. Ce critère est important pour notre projet parce que différents utilisateurs pourront modifier un même modèle. Ces utilisateurs pourront donc avoir besoin de savoir quelles sont les dernières modifications qui ont été apportées et par qui. Cette information sera d'autant plus précise si le système de gestion de versions utilisé peut traquer les changements ligne par ligne.

(4) *Ensemble de changements* de la catégorie *Fonctionnalités*. Ce critère spécifie si le répertoire supporte les ensembles de changements (façon de grouper un ensemble de modifications liées en un élément atomique). Cette fonctionnalité pourrait être intéressante pour notre projet. Par exemple, on pourrait vouloir regrouper dans un ensemble de changements l'ajout d'une compétence, l'ajout d'une production et le lien entre ces deux éléments du modèle de l'apprenant. Le support des ensembles de changements permettrait de voir cet exemple de modification

comme un ensemble : une compétence a été ajoutée et elle est illustrée par cette production, au lieu d'avoir à les considérer comme trois modifications différentes sans lien entre elles. Sans être critique pour notre projet, ce critère représente une plus-value intéressante.

Nous allons finalement présenter les critères de la dernière catégorie d'importance, soit les **critères intéressants**. Ils sont au nombre de cinq. Ces critères peuvent ajouter de la plus-value à notre système mais ils ne sont pas obligatoires.

(1) *Documentation disponible* de la catégorie *Information technique*. Ce critère décrit la qualité de la documentation disponible sur le logiciel. Comme nous allons adapter le logiciel à nos besoins et l'intégrer dans notre système, il est très intéressant pour nous qu'il existe une documentation de qualité pour nous aider dans notre tâche.

(2) *Statut de développement* de la catégorie *Information générale*. Ce critère décrit l'état courant de développement du logiciel, c'est-à-dire qu'il précise si le logiciel est en développement actif, uniquement maintenu, s'il est mis à jour, amélioré... Il est intéressant d'utiliser un outil en développement actif, cela montre que celui-ci est vivant et adapté aux besoins de ses utilisateurs (puisque modifié et mis à jour).

(3) *Signature des révisions* de la catégorie *Fonctionnalités*. Ce critère fait référence à l'intégration d'une signature digitale aux modifications. Ce critère peut être intéressant pour notre projet en ce qui concerne la certification, la validité des modifications. Toutefois, il existe d'autres méthodes pour assurer la certification des informations contenues dans le modèle (et avec différents niveaux de certifications). Particulièrement, une méthode basée sur les points de vue nous semble pertinente. Il est donc intéressant pour nous d'utiliser un outil de gestion de versions supportant la signature des révisions, mais ce n'est pas essentiel.

(4) *Tags* de la catégorie *Fonctionnalités*. Ce critère précise s'il est possible de donner des noms aux modifications. Il pourrait être intéressant de pouvoir nommer les modifications pour ensuite rechercher un type de modification particulier. Par exemple, si on nomme Ajout d'une compétence, les modifications qui consistent à

ajouter une compétence dans le modèle, il serait ensuite facile d'effectuer une recherche du type : quels sont les ajouts de compétences qui ont été effectués sur ce modèle ? Toutefois, il existe d'autres moyens d'effectuer ces recherches, en appliquant par exemple la recherche à l'objet informatique qui a été modifié. Pour notre projet, il peut donc être intéressant de pouvoir nommer les modifications, mais ce n'est pas un critère important.

(5) *Support international* de la catégorie *Fonctionnalités*. Ce critère précise si le logiciel a un support multi-langues. Ce critère n'est absolument pas critique mais si le logiciel qui correspond le mieux à nos besoins comporte de la documentation en français, tant mieux !

Le tableau suivant résume les critères présentés et les valeurs nécessaires ou souhaitées pour chacun d'eux.

Tableau II-4 Récapitulatif des critères et de leur valeur

Importance	Critère	Valeur
Discriminant	Licence	Doit être gratuite
	Licence libre	Oui
Déterminant	Modèle de l'accès simultané	<i>Merge</i>
	Langage de programmation	Doit inclure Java
	Portabilité	Excellent
	Définir des permissions	Oui
	Plateformes supportées	Doit inclure Windows
	Validation atomique	Oui
Important	Modèle du répertoire	Centralisé ou décentralisé
	Facilité de déploiement	Peu importe
	Historique du fichier ligne par ligne	Oui
	Ensemble de changements	Oui est mieux
Intéressant	Documentation disponible	Le plus possible
	Statut de développement	Le plus actuel possible
	Signature des révisions	Oui est mieux
	Tags	Oui est mieux
	Support international	Oui est mieux

2.3.1.1.2. Analyse comparative de logiciels de gestion de versions

Nous avons plus particulièrement effectué un comparatif des 16 outils sous licence libre que nous avons analysés selon les 17 critères présentés précédemment. Les tableaux présentant cette analyse se situent à l'appendice D page 182. Ils sont suivis des tableaux présentant la comparaison de tous les logiciels répertoriés dans notre recherche.

À la suite de cette analyse, nous avons retenu les deux systèmes les plus susceptibles de répondre aux besoins de notre projet : LibreSourceSynchronizer et Subversion (SVN). Le tableau suivant montre le comparatif de ces deux systèmes selon les 17 critères.

Tableau II-5 Comparatif des outils de gestion de versions

Catégorie	Critère	LibreSource Synchronizer	Subversion (SVN)
Information générale	Statut de développement	maintenu et nouvelles fonctionnalités en développement	développement actif
	Modèle du répertoire	client serveur	client serveur
	Modèle accès simultané	merge	merge or lock
	Licence	QPL	Apache-BSD
	Licence libre	oui	oui
	Portabilité	excellent	excellent
	Plateformes supportées	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	Java	C
	Documentation disponible	moyen	très bon
	Facilité de déploiement	excellent	Il faut Apache 2
Fonctionnalités	Validation atomique	oui	oui
	Définir des permissions	permission par répertoire ou branche	oui
	Ensemble de changements	partiel	partiel
	Historique du fichier ligne par ligne	oui	oui
	Signature des révisions	non	non
	Tags	oui	partiel
	Support international	?	oui

Suite à ce comparatif, nous avons choisi d'utiliser Subversion dans le cadre de notre thèse. En voici les raisons :

- Subversion est multi-plateformes et sous licence libre, et nous pourrions ainsi aisément l'utiliser;
- Subversion fonctionne sur le mode client/serveur et nous pensons qu'il est préférable dans notre recherche d'avoir un dépôt centralisé;
- Subversion semble être le nouveau CVS et en voie de devenir un des outils de gestion de versions les plus utilisés;
- Subversion dispose d'une API Java qui nous permettra facilement de l'utiliser avec notre modèle de l'apprenant.

Notre analyse de LibreSourceSynchronizer se trouve à l'appendice E page 207. Nous présenterons Subversion plus en détail dans la section d'implémentation du chapitre IV.

2.4. Conclusion

Dans ce chapitre, nous avons présenté une revue des écrits scientifiques concernant des concepts et approches technologiques que nous avons été amenés à utiliser dans le cadre de nos travaux de recherche. Plus précisément, nous avons étudié l'ePortfolio en tant que concept et en tant qu'outil. Nous proposons ainsi d'intégrer certaines caractéristiques de l'*ePortfolio* à notre modèle de l'apprenant. Principalement, notre modèle contiendra des productions de l'apprenant (comme le ferait un *ePortfolio*) qui pourront être reliées aux compétences. Ainsi, les compétences pourront être illustrées par des productions « concrètes » de l'apprenant. Enrichir le modèle de l'apprenant par des productions de celui-ci permet donc d'avoir d'éventuelles preuves, ou du moins, des illustrations et des exemples concrets de développement des compétences. De plus, un *ePortfolio* est témoin de l'évolution des apprentissages et du développement des compétences. Finalement, les *ePortfolios* sont des outils destinés à l'apprenant lui-même et sont visibles et modifiables par des acteurs humains.

Concernant la notion de contrat, nous avons vu qu'elle va nous permettre de gérer les interactions entre notre modèle de l'apprenant et d'autres systèmes informatiques. Nous allons nous inspirer des contrats créés en informatique mais nous n'avons pas implémenté cet aspect dans notre recherche. Nous nous sommes inspirée de cette notion pour concevoir une façon de gérer les interactions de notre système avec d'autres systèmes.

Ensuite, nous avons exploré la gestion de versions. Nous nous sommes attardée plus en détail sur ce point car cet aspect sera implémenté dans notre système. La

gestion de versions va nous être utile pour répondre à notre objectif visant à ce que notre modèle de l'apprenant évolue dans le temps au même titre que l'apprentissage et afin de soutenir l'apprentissage tout au long de la vie. Nous avons ainsi étudié les possibilités offertes par les logiciels de gestion de versions, possibilités que nous n'exploiterons pas intégralement. Par exemple, certains logiciels de gestion de versions gèrent les conflits pouvant exister entre différentes versions. Dans notre cas, comme nous avons différents acteurs qui pourront exprimer leurs opinions possiblement différentes elles aussi, nous gérerons les conflits à l'extérieur du système de gestion de versions. Après l'étude de divers logiciels selon plusieurs critères de sélection, nous avons choisi d'utiliser Subversion dans notre projet.

Dans le chapitre III, nous présentons notre proposition, soit un modèle cognitif de l'apprenant à points de vue multiples et évolutif. Nous allons expliquer en détail comment les éléments présentés dans les deux chapitres précédents ont été utilisés dans notre recherche.

Chapitre III - PROPOSITION ORIGINALE : UN MODELE DE L'APPRENANT COGNITIF, A POINTS DE VUE MULTIPLES ET EVOLUTIF

Dans ce chapitre, nous présentons le modèle conceptuel de notre proposition originale (section 3.1.) puis son opérationnalisation (section 3.2.). Dans le cadre de notre problématique et pour atteindre nos objectifs, nous proposons un modèle de l'apprenant cognitif, évolutif et à points de vue multiples. Ce modèle se construit au fur et à mesure des apprentissages. Dans la première section de ce chapitre, nous présentons les différents aspects du modèle conceptuel de notre proposition de modèle de l'apprenant, soit un modèle cognitif (section 3.1.1.), un modèle à points de vue multiples (section 3.1.2.) et un modèle évolutif (section 3.1.3.). Après nous expliquons comment ce modèle est opérationnalisé et comment il peut être utilisé (section 3.2.). Pour cela, nous introduisons une structure en trois couches utilisant les notions de métamodèle, de modèle et d'instance (section 3.2.1.). Nous précisons ensuite comment cette structure est exploitable et mise en œuvre pour permettre l'utilisation de notre proposition dans des contextes variés (sections 3.2.2., 3.2.3. et 3.2.4.). Nous finissons en présentant une vue d'ensemble de l'architecture globale de notre proposition (section 3.2.5.). Ce chapitre présente le résultat de l'étape 2.2 *Conception*, de la phase 2 *Élaboration et construction* de notre démarche de recherche.

3.1. Notre proposition : modèle conceptuel

3.1.1. Modèle cognitif de l'apprenant

Le modèle de l'apprenant que nous proposons est un modèle cognitif, c'est-à-dire qu'il permet de représenter l'état cognitif d'un apprenant à un moment donné. Ce n'est pas un modèle vraisemblable cognitivement, c'est-à-dire qu'il ne reproduit pas le fonctionnement cognitif du cerveau de l'apprenant. Pour modéliser l'état cognitif des apprenants, nous avons choisi une approche par compétences parce que celle-ci nous permet d'avoir un référencement sémantique riche. Nous allons aussi intégrer certaines caractéristiques de l'*ePortfolio* dans notre modèle ainsi que des informations personnelles et professionnelles (IPP) concernant l'apprenant représenté. Les sections suivantes visent à présenter ces trois composantes de notre modèle de l'apprenant.

3.1.1.1. Compétences

Les compétences se situent au cœur de notre modèle de l'apprenant. Nous avons expliqué dans la conclusion du chapitre I pourquoi nous avons choisi d'adopter une approche par compétences pour modéliser l'état cognitif d'un apprenant.

Comme nous l'avons précisé dans le cadre conceptuel, nous allons travailler avec la définition proposée par Paquette (2002a) et présentée à la section 1.2. intitulée "L'approche par compétences" (page 30) : le concept de compétence se définit comme « la relation entre une habileté générique appliquée à une connaissance à un certain niveau de performance ». Le modèle conceptuel de compétence tel qu'utilisé dans notre projet est tiré de la définition de Paquette et présente des modifications par rapport à son ontologie du concept de compétence (présentée au chapitre I). Avant de présenter notre modèle conceptuel de compétence, nous allons (re)présenter un schéma exposant la proposition de Paquette afin de montrer quels éléments nous conservons et quels sont ceux que nous ajoutons. Selon l'ontologie de Paquette

(2007), une compétence dispose d'un contexte d'utilisation (compétence pré-requise, compétence actuelle ou compétence cible), elle a un énoncé en langage naturel et elle peut permettre d'annoter une ressource (ressource dans son sens large, incluant les matériels pédagogiques mais aussi les acteurs). La compétence est aussi liée à une connaissance tirée d'une ontologie de domaine ainsi qu'à une habileté générique qui, elle-même, est liée à des indicateurs de performance. La figure III-1 représente cette proposition de compétence adaptée à notre projet de recherche.

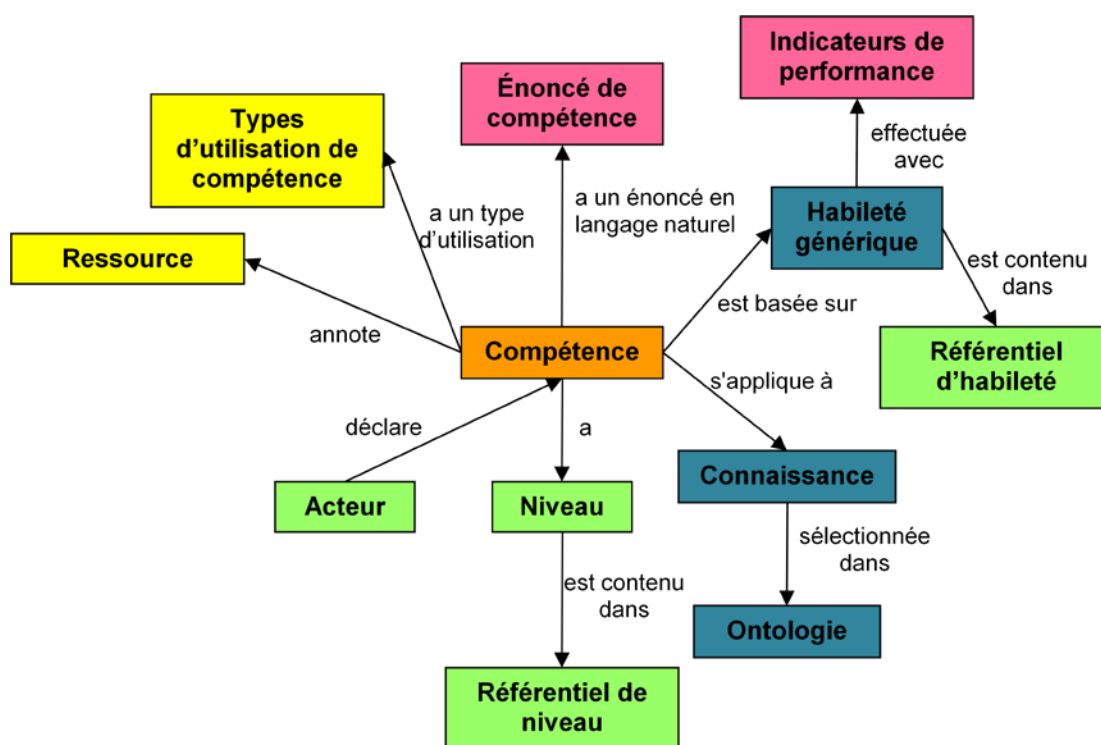


Figure III-1. Le concept de compétence de Paquette (2007) adapté

Les éléments en jaune ne sont pas conservés dans notre proposition. Nous ne souhaitons pas spécifier le type d'utilisation (c'est-à-dire dans la proposition de Paquette, compétence pré-requise, actuelle ou cible) des compétences contenues dans les modèles d'apprenant. Toutes les compétences contenues dans un modèle sont possédées par l'acteur qui possède le modèle en question. De même, nous n'avons pas besoin d'utiliser les compétences pour annoter des ressources, comme le permet la proposition de Paquette (2007). Par contre, dans le cas où la ressource est un

acteur, alors cela signifie que l'acteur possède la compétence. Pour ce cas particulier de la proposition de Paquette, le lien entre la compétence et l'acteur qui la possède est implicite dans notre modèle d'apprenant puisque les compétences contenues dans un modèle sont toutes possédées par l'acteur en question.

Les éléments en bleu sont ceux que nous conservons dans notre proposition (habileté et connaissance liée à une ontologie). Les éléments en rose sont conservés dans notre proposition mais représentés différemment. Dans notre proposition, une compétence dispose d'un énoncé en langage naturel qui est représenté sous forme d'attribut. Et les indicateurs de performances se retrouvent dans le Niveau (comme nous l'expliquerons plus en détail dans la suite de cette section). Finalement, les éléments verts sont des ajouts de notre part. Le niveau remplace les indicateurs de performance de Paquette et il est lié à un référentiel de niveaux. De même, l'habileté est, elle aussi, liée à un référentiel d'habiletés. Nous avons aussi ajouté un acteur. Nous allons maintenant expliquer plus en détail les éléments constituant notre proposition. La figure III-2 représente cette proposition du modèle conceptuel du concept de compétence tel qu'utilisé dans notre modèle de l'apprenant. Cette figure représente notre modèle de l'apprenant constitué à cette étape uniquement du concept de compétence et que nous appelons donc modèle de l'apprenant à l'étape 1.

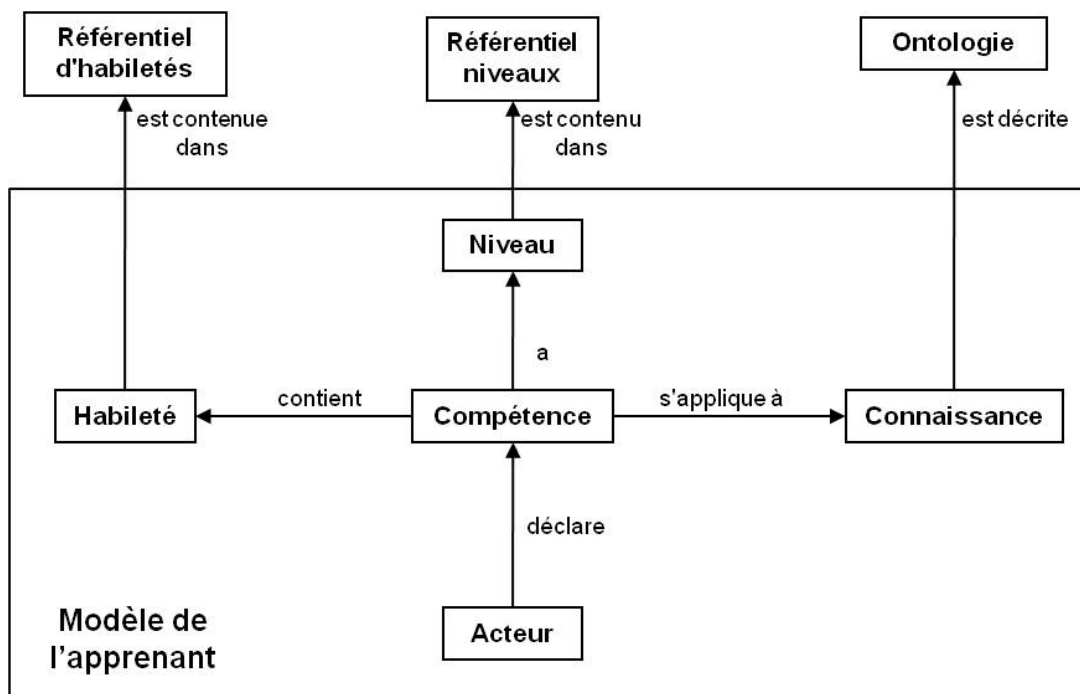


Figure III-2. Modèle conceptuel du modèle cognitif de l'apprenant – étape 1

Ainsi, dans notre modèle, une compétence est définie selon quatre caractéristiques : un acteur, une connaissance, une habileté et un niveau. Nous allons présenter ces caractéristiques plus en détail dans les paragraphes qui suivent.

Nous avons aussi ajouté des attributs à une compétence. Une compétence a un nom, est décrite textuellement (équivalent de l'énoncé de compétence de Paquette), dispose d'une date d'ajout (qui permet de savoir quand la compétence a été ajoutée dans le modèle de l'apprenant) et un contexte (contexte dans lequel la compétence a été déclarée et ajoutée au modèle de l'apprenant). Le contexte peut être un cours, une expérience professionnelle... Le contexte est aussi décrit par une catégorie. La catégorie 1 regroupe les contextes académiques dans lesquels des acteurs humains interviennent, soit principalement les cours universitaires donnés sur campus. La catégorie 2 regroupe les contextes non académiques, comme par exemple le cas où un apprenant décide de faire un bilan de compétences. La catégorie 3 concerne les cours en ligne dans lesquels un système d'apprentissage gère au moins une partie des apprentissages et des évaluations.

Nous avons déjà dit que le lien entre une compétence et l'acteur qui la possède est implicite dans notre modèle (les compétences contenues dans un modèle sont possédées par le détenteur du modèle, soit la personne décrite, représentée par le modèle. Ainsi, dans notre modèle de l'apprenant, l'**acteur** lié à la compétence est plutôt celui qui l'a déclarée (celui qui déclare que l'apprenant possède telle compétence). Cette information permet de savoir en tout temps quelle est la source de la compétence, qui a ajouté la compétence dans le modèle (l'apprenant lui-même, un de ses professeurs, un pair...). L'acteur peut être une personne physique ou non (comme une institution par exemple).

Les trois autres caractéristiques sont liées : une compétence est une habileté appliquée à une connaissance à un certain niveau de maîtrise. La **connaissance** est tirée d'un domaine disciplinaire ou transversal. Elle est ainsi liée à l'ontologie représentant ce domaine. L'ontologie n'est pas incluse dans notre modèle, notre modèle contient plutôt un lien vers l'ontologie. L'**habileté** est quant à elle liée à un référentiel qui se situe lui aussi à l'extérieur de notre modèle. De plus, une habileté a un nom et un rang qui représente sa position dans le référentiel. En effet, les référentiels classifient les habiletés. Par exemple, le référentiel de Paquette (2002a) contient 10 habiletés (voir chapitre I page 32), où la cinquième est *appliquer* et la huitième *synthétiser*. Dans notre modèle, ces habiletés auront donc comme rang respectif cinq et huit. Le concept de **niveau** correspond aux indicateurs de performance de Paquette (présentés dans le chapitre I). Nous utilisons le terme niveau pour niveau d'atteinte ou de maîtrise. C'est le niveau d'atteinte de l'application de l'habileté sur la connaissance (à quel niveau l'acteur concerné peut-il exercer l'habileté sur la connaissance). Un niveau dispose d'un nom, d'un lien avec un référentiel (qui est à l'extérieur de notre modèle) et d'un rang, comme pour l'habileté. Les liens avec un référentiel d'habiletés, un référentiel de niveaux et avec une ontologie représentent le référencement sémantique que nous pensons indispensable à une adaptation des apprentissages pertinente aux besoins et caractéristiques des apprenants.

Nous illustrons notre modèle de l'apprenant avec les référentiels (d'habiletés et de niveaux, soit les indicateurs de performance) de Paquette parce que nous fondons notre travail sur les travaux de Paquette (2002a, 2002b, 2007) sur la compétence et son utilisation en ingénierie pédagogique. Toutefois, notre projet s'inscrit dans une volonté de compatibilité de notre modèle avec différents contextes. Le modèle conceptuel de compétence que nous proposons est suffisamment souple pour cette ouverture à différents contextes d'utilisation. Premièrement, le fait de disposer de référentiels (d'habiletés ou de niveaux) à l'extérieur du modèle de l'apprenant permet d'utiliser le modèle avec différents référentiels et donc dans différents contextes. Aussi, il est possible de ne définir la compétence que textuellement, sans préciser les champs connaissance et habileté. Dans tous ces cas d'utilisations différentes du concept de compétence, le modèle perd de sa richesse, notamment au niveau du référencement sémantique. En effet, si différents référentiels sont utilisés (que ce soit des référentiels d'habiletés ou de niveaux), la comparaison des compétences ne pourra plus se faire automatiquement. En effet, pour rendre cette comparaison possible, il faudra préalablement comparer les différents référentiels et créer un traducteur permettant de passer de l'un à l'autre ou créer une échelle de comparaison entre les deux. Si la compétence n'est définie que textuellement (par une description) et que les champs connaissance et habileté ne sont pas renseignés, alors toute comparaison est impossible, et il n'y a plus aucun référencement sémantique. Le modèle de l'apprenant devient une liste d'énoncés de compétences, et non une liste de compétences référencées sémantiquement. Permettre ce type d'utilisation rend le modèle de l'apprenant moins puissant mais plus « utilisable » dans le sens qu'il pourra être utilisé dans des contextes variés selon leur vision du concept de compétence, ce qui peut être considéré comme un enrichissement du modèle. Nous illustrerons ces propos dans le chapitre V au cours de simulations qui s'effectueront avec différentes utilisations du concept de compétence et avec différents référentiels.

3.1.1.2. Productions

Comme nous l'avons vu dans la section 2.1. *ePortfolio* du chapitre II (page 39), l'*ePortfolio* est une nouvelle façon de collecter des informations sur un apprenant. Celui-ci contient les productions de l'apprenant. Selon notre étude sur les *ePortfolios* (Moulet, 2006a), présentée dans l'état de la question, notre travail est orienté vers les *ePortfolios* personnels d'apprentissage (utilisés pour l'autoréflexion, la reconnaissance de ses propres habiletés ainsi que pour montrer l'apprentissage.) Ces *ePortfolios* fournissent un cadre pour évaluer les progrès d'apprentissage et permettent de démontrer comment les habiletés se développent dans le temps. Toutefois, il serait possible d'utiliser le modèle de l'apprenant dans un contexte professionnel, par exemple, en présentant le modèle comme un CV professionnel électronique, ou en analysant le modèle afin de prendre des décisions concernant sa carrière. Nous avons effectué une étude des outils *ePortfolio* existants. Principalement, nous avons étudié l'outil l'*Open Source Portfolio* (OSP) (Moulet, 2006b). Cet outil se trouvait finalement assez loin des préoccupations de notre projet. C'est pourquoi nous avons choisi de ne pas intégrer un outil *ePortfolio* existant mais plutôt de nous inspirer de ce type d'outil pour intégrer à notre projet les fonctionnalités pertinentes selon nos objectifs.

Nous nous sommes donc inspirés de la notion d'*ePortfolio* pour créer notre modèle de l'apprenant. Plus précisément, les caractéristiques des *ePortfolios* que nous avons retenues sont les productions des apprenants, le fait que le modèle soit visible et modifiable par des acteurs humains et la possibilité d'observer l'évolution de ses apprentissages. L'intégration dans notre modèle de ces concepts associés à l'*ePortfolio* va permettre aux apprenants de réfléchir sur leurs apprentissages et d'être ainsi mieux engagés dans leur formation.

Les productions peuvent être utilisées pour illustrer les compétences, c'est-à-dire pour leur servir d'exemples ou de preuves. Notre modèle conceptuel de modèle de l'apprenant contient ainsi un concept production lié au concept compétence. La figure III-3 représente notre modèle conceptuel à l'étape 2.

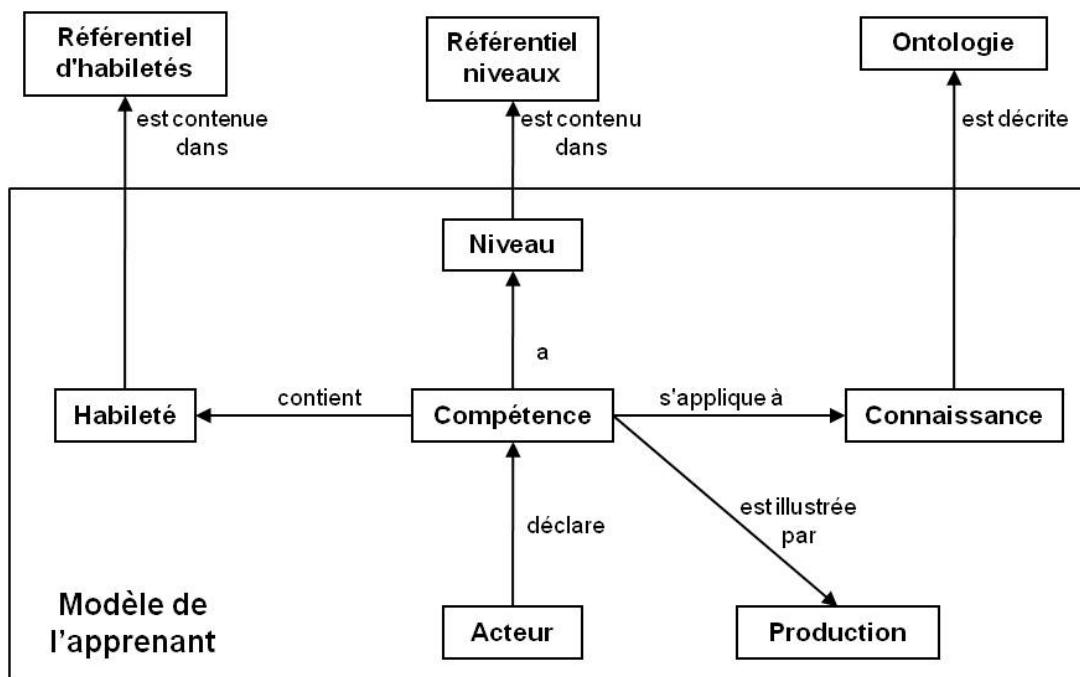


Figure III-3. Modèle conceptuel du modèle cognitif de l'apprenant – étape 2

Dans notre modèle, une production a un nom, une description textuelle, une date de création, une date d'ajout (dans le modèle), un lien vers son emplacement, un contexte (celui de son ajout au modèle) et éventuellement un tampon (ce tampon est apposé par un acteur pour signifier que la production a de la valeur, par exemple, un professeur peut tamponner les trois meilleures productions de la session universitaire). Notre modèle ne contient donc pas la production « physique » mais plutôt une sorte de fiche de métadonnées sur cette production contenant l'adresse physique de la production. De plus, notre modèle de l'apprenant contient un ensemble de liens entre les compétences et les productions contenues dans le modèle. Ces liens sont documentés eux aussi par un élément de contexte, ainsi que par deux acteurs. Le premier acteur (obligatoire) est l'auteur du lien, cette information permet de savoir en tout temps l'origine du lien. Le second est un valideur (optionnel) qui donne de la valeur à la signification du lien, qui certifie que le lien est juste.

Une même production peut documenter plusieurs compétences comme une compétence peut être illustrée par plusieurs productions. En effet, lors de la création

d'une production, plusieurs compétences peuvent être mises en œuvre et une même compétence peut être mobilisée lors de la construction de différentes productions. Une compétence peut n'être illustrée par aucune production et une production peut faire partie du modèle de l'apprenant sans être liée à aucune compétence, ces liens pouvant être ajoutés après (si jugé nécessaire et utile par les acteurs concernés).

3.1.1.3. Informations personnelles et professionnelles

Les informations personnelles et professionnelles (IPP) créent le lien entre l'apprenant et son monde (le contexte dans lequel il évolue, contexte social, culturel, professionnel...). L'apprentissage est un processus continu qui se produit dans un contexte spécifique, mais qui peut varier. En effet, l'apprentissage a lieu dans des milieux scolaires, mais aussi dans des environnements professionnels. Les IPP vont nous permettre de recueillir des informations sur les apprenants, en dehors des contextes de formation, plus spécifiquement, elles vont nous permettre de recueillir les compétences développées dans des contextes professionnels et personnels. Ce type d'information peut être particulièrement pertinent dans le cas de l'apprentissage continu. Comme pour l'*ePortfolio*, nous avons étudié la documentation scientifique avant de concevoir cet élément de notre modèle. Nous avons identifié différents projets que nous avons étudiés et desquels nous nous sommes inspirés (vCard du Internet Mail Consortium, 1998 et HR-XML de HR-XML Consortium, 2007). Dans notre modèle de l'apprenant, les IPP comportent principalement des informations sur les expériences professionnelles qui peuvent illustrer les compétences (voir la figure III-4).

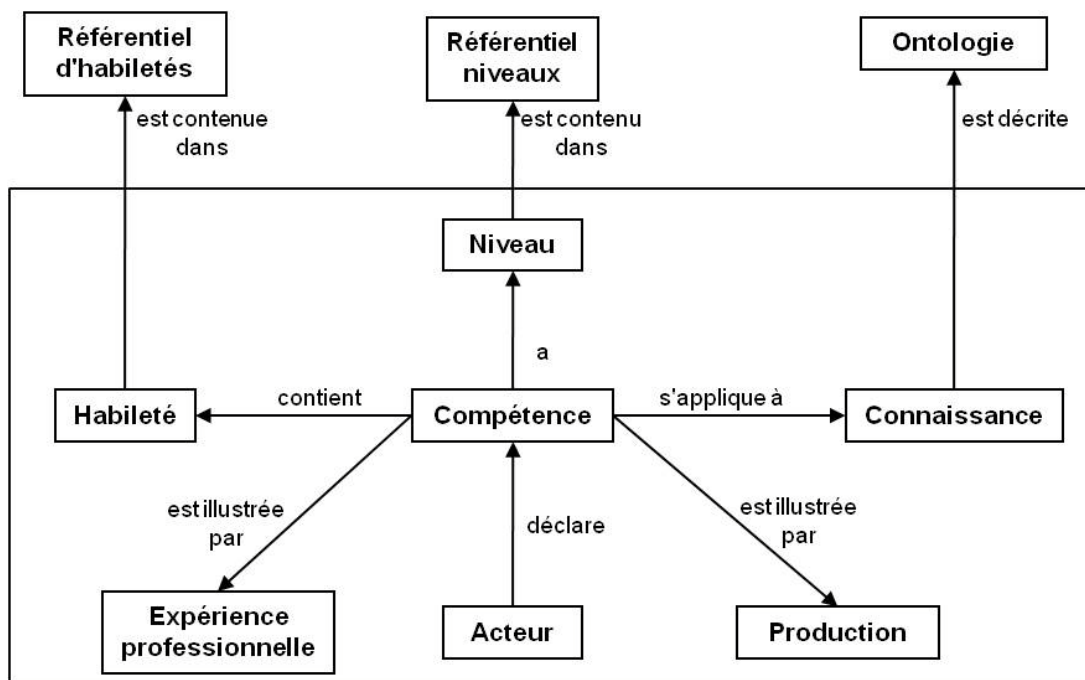


Figure III-4. Modèle conceptuel du modèle cognitif de l'apprenant

Les expériences professionnelles sont décrites par une date de début, une date de fin, une date d'ajout dans le modèle, une description textuelle et un nom d'entreprise. De plus, le modèle de l'apprenant que nous proposons contient un ensemble de liens entre des compétences et des expériences professionnelles qui expriment le fait qu'une expérience permet de documenter une compétence. En effet, un acteur mobilise des compétences au cours de ses expériences professionnelles. Ces expériences professionnelles peuvent ainsi illustrer les compétences, leur servir de preuve. Comme pour les liens entre les compétences et les productions, ces liens sont décrits par le contexte au cours duquel ils ont été ajoutés et par deux acteurs, celui ayant ajouté le lien (cette information est automatiquement ajoutée au modèle de l'apprenant) et un le certifiant. L'acteur certifiant le lien n'est pas toujours présent puisque les liens ne sont pas obligatoirement validés ou certifiés. Comme pour la production, l'acteur certifiant les liens (entre compétence et production ou information professionnelle) peut être vu comme un tampon sur le lien qui permet de certifier que le lien est valable. En effet, les acteurs faisant confiance à l'acteur qui a

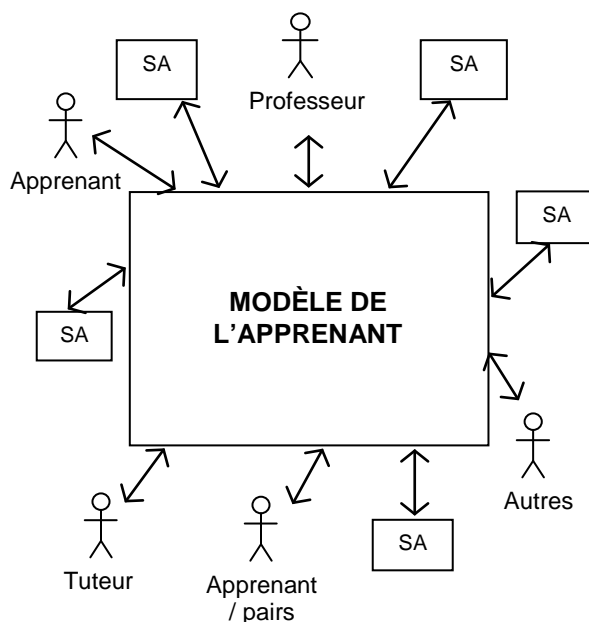
tamponné le lien auront confiance en ce lien et seront assurés que la production est bien un exemple d'utilisation de la compétence visée par le lien (ou que l'expérience professionnelle a bien permis à l'acteur de démontrer qu'il possède la compétence visée par le lien).

3.1.2. Modèle de l'apprenant à points de vue multiples

Dans cette recherche, nous appréhendons l'apprentissage comme étant un processus de construction des connaissances résultant de l'interaction entre un apprenant et son environnement (De Vries et Baillé, 2006). C'est de cet environnement que l'apprenant retire ses conditions d'apprentissage. Ainsi, nous pouvons considérer l'apprentissage comme étant un acte social qui évolue en tenant compte des interactions avec l'environnement (Vygotski, 1934-1997). Ceci fait de l'apprentissage un processus dynamique. Nous avons présenté une vue isolée du modèle de l'apprenant, pourtant, celui-ci est situé dans un contexte social riche. Comme la figure III-5 l'illustre, l'apprentissage est un processus social engageant différents acteurs (acteurs pouvant interagir avec le modèle (seulement en consultation par exemple), à ne pas confondre avec les acteurs qui déclare une compétences). Nous avons identifié cinq rôles : l'apprenant, ses pairs, les professeurs, les tuteurs et les autres (comme par exemple un membre de l'équipe administrative ou un responsable de stage du milieu industriel). Ces différents acteurs interagissent de façon différente avec le modèle de l'apprenant et peuvent effectuer différentes tâches (consulter, ajouter, modifier, supprimer ou certifier) sur différents éléments du modèle (productions, compétences, liens entre des éléments, etc.).

Hansen et McCalla (2003) affirment qu'ouvrir le modèle de l'apprenant aux apprenants et à leurs pairs permet d'aider la réflexion sur les apprentissages. Devedzic (2006) précise qu'en ouvrant le modèle de l'apprenant, celui-ci est conscient de ce que le système « pense » de ses connaissances, ses progrès, ses besoins et qu'il peut ainsi réfléchir sur ces données et contrôler ses apprentissages. De plus, l'apprenant peut

aussi ajuster certaines données contenues dans son modèle afin de permettre une personnalisation plus adaptée. Nous pensons aussi qu'il est important d'ouvrir le modèle de l'apprenant d'une part à lui-même pour favoriser la réflexion sur ses apprentissages, mais aussi aux différents types d'acteurs intervenant dans le processus d'apprentissage.



SA : système d'apprentissage

Figure III-5. Modèle de l'apprenant et son contexte social

Nous avons déjà précisé que l'apprentissage a lieu dans différentes situations dans lesquelles l'apprenant se trouve et par ses interactions sociales. Au cours de ses apprentissages, l'apprenant interagit avec différents systèmes d'apprentissage ainsi que différents acteurs. Ces systèmes d'apprentissage et ces acteurs devraient pouvoir enregistrer les résultats de ces interactions dans le modèle de l'apprenant. Le modèle est ainsi vu de différents points de vue. Comme nous l'avons précisé dans la section traitant des points de vue du chapitre I (section 1.3. page 34), différents points de vue sur un même objet peuvent être consensuels ou conflictuels (Dieng-Kuntz *et al.*, 2000). Un point de vue consensuel apparaît lorsque deux ou plusieurs observateurs regardant les mêmes attributs d'un objet y voient les mêmes valeurs. Dans le contexte

de notre thèse, des points de vue consensuels signifient que si deux acteurs ou systèmes d'apprentissage ont des informations à fournir sur une compétence ou sur un lien entre une compétence et une production d'un apprenant à un moment particulier, ces informations seront les mêmes : les acteurs sont d'accord sur la déclaration d'une des compétences de l'apprenant. Au contraire, des points de vue conflictuels apparaissent lorsque deux observateurs regardant un même objet au même moment voient des valeurs différentes pour certains attributs. Dans le cas d'un modèle de l'apprenant orienté compétences, nous devons nous attendre à voir émerger des conflits relativement souvent. En effet, l'évaluation d'une compétence est une tâche complexe. Dieng-Kuntz *et al.* identifient deux autres notions : la correspondance et le contraste qui font référence à des différences sur la notation sémantique des concepts. Ces deux notions ne nous concernent pas car soit elles sont résolues par l'utilisation d'un référentiel de compétences commun, soit la situation est un conflit clair. Ces deux notions finissent ainsi par être équivalentes à une des deux catégories que nous avons retenues : consensus et conflit.

Pour savoir si des compétences sont évaluées de façon conflictuelle ou consensuelle, il faut les comparer. La comparaison des compétences est une tâche qui peut s'avérer être difficile. Des compétences peuvent différer selon l'habileté générique ou le niveau qui leur sont associés. Pour savoir si deux compétences sont équivalentes ou non, il faut donc comparer ces éléments. On peut aussi vouloir comparer des compétences pour les « classer ». Par exemple, si un étudiant peut appliquer une connaissance, on peut en déduire qu'il est capable d'en parler. La façon de comparer et d'éventuellement classer les compétences dépend des référentiels (d'habiletés et de niveaux) utilisés. Par exemple, le référentiel d'habiletés génériques de Paquette ordonne celles-ci, ainsi les habiletés inférieures sont incluses dans les habiletés supérieures (l'habileté 1 est incluse dans l'habileté 2 qui est incluse dans l'habileté 3, etc.). Cette règle sert de base à la comparaison de compétences définies avec ce référentiel.

Nous proposons ainsi un modèle de l'apprenant à points de vue multiples. Cette caractéristique de notre proposition rejoint un des points que nous avons retenus à la suite de notre étude de l'outil *ePortfolio*, le fait que ces outils soient accessibles par différents acteurs humains. Nous enrichissons cette possibilité en offrant la possibilité aux acteurs machine de disposer d'un point de vue. Notre modèle de l'apprenant est un modèle à points de vue multiples composé d'un **point de vue consensus** et de **points de vue associés** (voir la figure III-6).

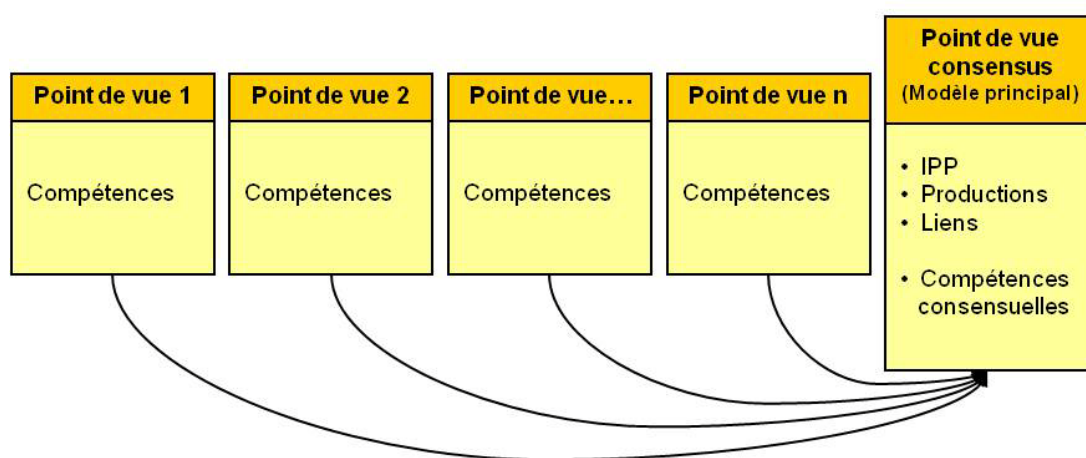


Figure III-6. Modèle de l'apprenant à points de vue multiples

Le point de vue consensus correspond à l'intersection des points de vue, c'est-à-dire qu'il contient toutes les informations sur l'apprenant qui font consensus auprès des différents acteurs humains et machines. Ces informations incluent les informations personnelles et professionnelles (IPP) ainsi que les productions de l'apprenant. Concernant les compétences, seulement les compétences acceptées par tous les acteurs sont conservées dans le point de vue consensus, c'est-à-dire les compétences consensuelles. Pour les liens (liens entre compétences et productions et entre compétences et IPP), ceux-ci sont ajoutés au point de vue consensus sans que cela ne signifie obligatoirement qu'un consensus a été obtenu entre tous les acteurs ayant un point de vue. Pour tester la validité d'un lien, un autre moyen que l'obtention de consensus est utilisé, c'est celui de certification par une autre personne que celle qui a ajouté le lien (toutefois, la personne certifiant un lien doit elle aussi

avoir un point de vue et donc accès au modèle de l'apprenant concerné). Comme nous l'avons dit, les liens peuvent être documentés par un acteur qui valide le lien. Cela donne de la valeur à la crédibilité du lien, cela permet de le valider (si bien sûr on fait confiance à l'acteur qui l'a certifié). Le point de vue consensus peut aussi être vu comme étant le modèle de l'apprenant principal ou consensus auquel sont reliés différents points de vue appartenant à différents acteurs. Dans le reste du texte, pour éviter la confusion, nous parlerons de point de vue consensus.

Sont liés à ce point de vue consensus autant de points de vue qu'il y a d'acteurs différents autorisés à modifier le modèle (tuteurs, professeurs, pairs, systèmes d'apprentissage...). Ces points de vue ne sont pas pré-établis, ils sont créés en fonction du contexte d'apprentissage de l'apprenant. Il peut y avoir un point de vue par type d'acteur, c'est-à-dire par rôle (un point de vue professeur, un point de vue pair, un point de vue système d'apprentissage en ligne...), ou alors un point de vue par individu (dans ce cas, il y aura plusieurs points de vue professeur, plusieurs points de vue pair...). Cela dépendra du contexte d'utilisation et de ses contraintes. En effet, une université peut, par exemple, être intéressée à donner un point de vue par professeur enseignant à l'apprenant alors qu'une autre peut préférer donner un seul point de vue pour l'ensemble des professeurs enseignant à cet apprenant. Cet élément, qu'on appellera « nombre de points de vue », devra être précisé selon les contextes d'utilisation, il pourra être précisé dans les contrats (ils permettent de gérer les interactions entre le modèle et les systèmes d'apprentissage, voir section 3.2.2.1. Interactions par l'utilisation de contrats à la page 85) et pourra changer au cours du temps selon l'utilisation qui est faite d'une instance du modèle de l'apprenant.

Il est important de noter ici qu'un acteur peut jouer différents rôles et être ainsi associé à différents points de vue. Par exemple, un professeur peut suivre un cours et jouer alors le rôle d'apprenant. Toutefois, lors de l'utilisation d'un modèle d'apprenant à un moment particulier, l'acteur doit choisir quel rôle il joue dans ce cas d'utilisation spécifique. Autrement dit, lorsqu'un acteur se connecte à un modèle de

l'apprenant particulier, il le fait selon un et un seul rôle. Par contre, ce même acteur peut jouer un rôle différent sur le modèle d'un autre apprenant.

Lorsqu'une modification concernant les compétences est proposée par un des acteurs, le point de vue correspondant à cet acteur est modifié et mis à jour. Nous proposons l'utilisation d'une stratégie de négociation pour permettre de gérer les différences qui pourraient y avoir entre les différents points de vue. Plusieurs stratégies de négociation peuvent être définies et appliquées selon les contextes d'utilisation du modèle de l'apprenant. Un exemple de stratégie serait que si une compétence est déclarée par un professeur et un apprenant, c'est la déclaration du professeur qui est conservée. Nous verrons dans la suite de ce chapitre quand et comment la stratégie de négociation intervient au cours de l'utilisation du modèle de l'apprenant.

Il faut aussi soulever la question de confidentialité en ce qui concerne les points de vue. Le point de vue consensus est visible par tous les utilisateurs ayant accès au modèle; par contre, les points de vue peuvent avoir un accès limité. Par exemple, on peut choisir de rendre le point de vue des acteurs du type professeur visible uniquement par les acteurs professeur et tuteur. Pour les points de vue d'individu, ils peuvent être définis comme privé ou public. Plus précisément, chaque information contenue dans un point de vue individuel peut être rendue publique ou gardée privée, au choix de l'individu dont c'est le point de vue. Cet aspect n'a pas été abordé dans la thèse.

3.1.3. Modèle de l'apprenant évolutif

L'apprentissage est un processus dynamique. Un modèle de l'apprenant doit refléter cette caractéristique de l'apprentissage et il doit ainsi évoluer. Un modèle de l'apprenant évolutif se met à jour en fonction des progrès réalisés au cours des apprentissages. McCalla, Vassileva, Greer et Bull (2000) proposent, eux aussi, un modèle de l'apprenant dynamique dans leur approche de modélisation de l'apprenant

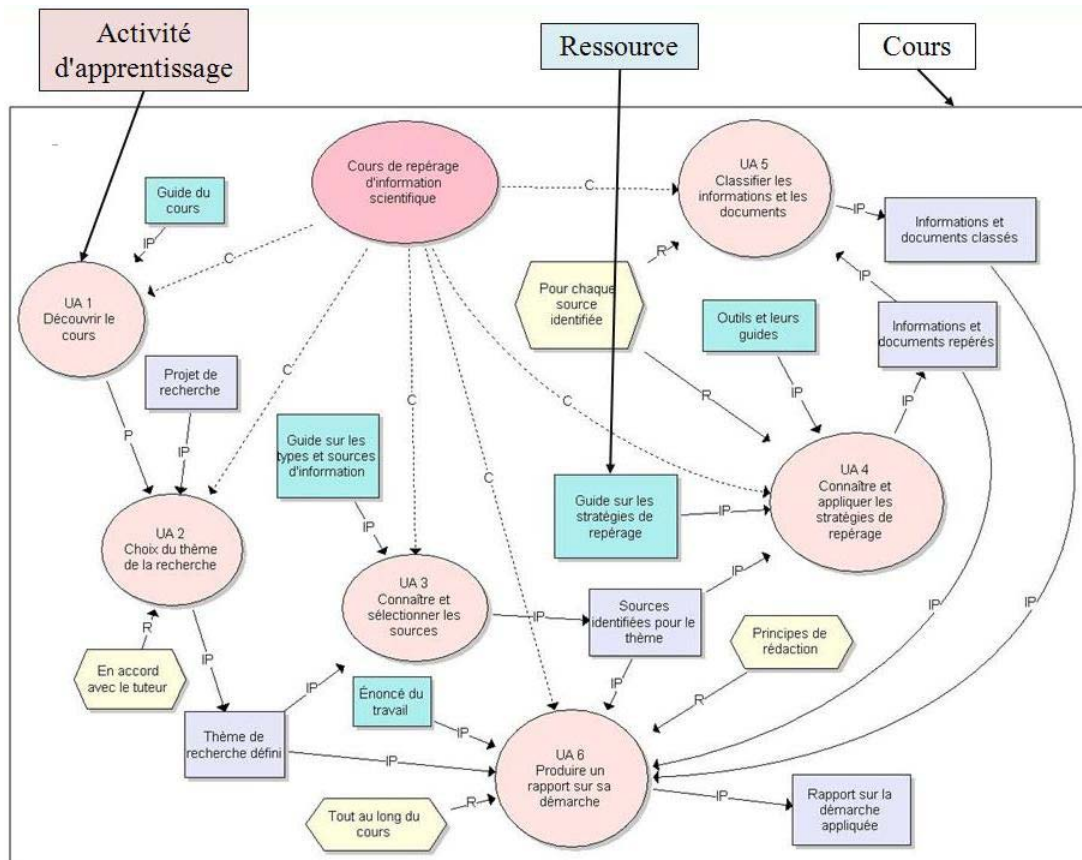
active et ouverte. Dans leur projet, le modèle de l'apprenant est calculé juste-à-temps pour un besoin particulier. Notre approche diffère de la leur parce que nous proposons un modèle global, contenant toutes les informations recueillies sur l'apprenant depuis la création du modèle. Deux aspects sont à considérer concernant l'évolution du modèle de l'apprenant dans le temps : d'une part, la nature des changements apportés que nous appelons modifications et d'autre part l'évolution à proprement parler, soit la nouvelle version du modèle (due à une ou des modifications), que nous appelons évolution ou version dans le reste du document. Nous pouvons nous poser quatre questions concernant l'aspect évolutif de notre modèle de l'apprenant, les deux premières concernent les modifications, la troisième les modifications et les versions et la dernière uniquement les versions :

- (1) quelle est la nature des modifications, c'est-à-dire quelles informations sont modifiées ?
- (2) qui peut effectuer une modification ?
- (3) quand une modification peut-elle avoir lieu ? quand une nouvelle version est créée ?
- (4) qu'est-ce qui évolue ? le modèle complet ? une partie du modèle uniquement ? quels éléments du modèle de l'apprenant sont contenus dans une nouvelle version créée à la suite des modifications ?

Concernant la première question, nous avons identifié cinq types de modification : 1) une production de l'apprenant, 2) une compétence, 3) un lien entre une compétence et une IPP ou une production, 4) une production, une compétence ainsi que le lien entre les deux et 5) des traces de l'interaction de l'apprenant avec le système d'apprentissage en ligne. Nous ne considérons pas les IPP car ces informations ne peuvent pas évoluer dans le temps, dans le sens d'évolution des apprentissages. Elles peuvent changer, être modifiées (une personne peut changer d'adresse par exemple), mais ces changements ne représentent pas une évolution des apprentissages. La structure de notre modèle de l'apprenant peut facilement contenir les quatre premiers types d'information. En ce qui concerne les traces d'interaction avec le système, seules celles concernant les éléments de notre modèle seront prises

en considération. Les autres types de traces d'interaction (comme la durée, le nombre de connexions au système...) ne seront pas prises en compte dans notre thèse mais elles pourraient constituer un élément supplémentaire éventuellement ajouté au modèle si besoin est. Dans ce cas, une analyse de la pertinence des données à recueillir devra être réalisée (il n'est pas nécessaire de recueillir une quantité d'information trop grande si le système n'est pas en mesure de les analyser). De plus, une analyse devra aussi être effectuée pour interpréter les traces recueillies et pouvoir ainsi les exploiter pertinemment. Ce type de travail concernant le recueil et l'analyse des traces d'interaction représente un projet à part entière et ne rentre pas dans notre problématique, c'est pourquoi nous ne nous en préoccupons pas spécifiquement.

Concernant la question *qui peut effectuer une modification ?*, deux types d'acteur peuvent effectuer une modification : le système d'apprentissage et les acteurs humains. Comme pour la visibilité des points de vue, des droits pourront être concédés aux acteurs quant aux modifications qu'ils peuvent effectuer sur le modèle. Par exemple, les professeurs pourraient modifier les compétences et les liens mais ne pourraient pas ajouter de production. Nous parlerons de cet aspect plus en détail dans la section concernant l'utilisation du modèle. Pour la question *quand une modification peut-elle avoir lieu ?*, dans le cas où le modèle de l'apprenant est branché à un système d'apprentissage, celui-ci peut effectuer une modification ou déclencher une version à quatre niveaux de granularité. Le modèle peut évoluer à la fin d'une formation, à la fin d'un cours, à la fin d'une activité d'apprentissage (éventuellement incluse dans un cours) et après l'utilisation d'une ressource (voir la figure III-7).



UA : unité d'apprentissage

Figure III-7. Exemple d'un cours et des évènements qui peuvent générer une évolution du modèle de l'apprenant

La figure a été réalisée avec le logiciel MOT élaboré au centre de recherche LICEF (Paquette, 2002a). La légende du langage MOT se trouve à l'appendice F page 211.

Le choix d'utiliser l'un ou l'autre des niveaux de granularité peut dépendre de contraintes institutionnelles. Par exemple, à l'université, si un étudiant possède les compétences visées d'un module particulier d'un cours, il devra tout de même suivre le cours dans son ensemble car actuellement, les inscriptions à des cours universitaires ne permettent pas de suivre une partie du cours uniquement. Au contraire, dans le monde professionnel, les employés peuvent suivre uniquement les modules dont ils ont besoin. Les acteurs machines peuvent aussi effectuer une modification et ce à n'importe quel moment. Pour tous les types d'acteurs, une nouvelle version est créée à chaque fois qu'un acteur sauvegarde ses modifications.

Tous les acteurs ont la même version du modèle pour lequel un historique de modifications existe, précisant la nature et l'auteur de chacune de ces modifications.

La dernière question, *quels éléments du modèle de l'apprenant sont contenus dans une nouvelle version créée à la suite d'une évolution ?*, est importante à considérer. Dans notre projet, tous les éléments du modèle sont conservés dans chaque évolution sauf les productions (le fichier de la production). Chaque version du modèle contient ainsi le modèle de l'apprenant principal et ses points de vue associés, c'est-à-dire que chaque version contient l'ensemble des compétences, les fiches descriptives des productions (contenant un lien vers le fichier qui est la production à proprement parler et qui est commun à l'ensemble des versions), les liens entre compétence et production et entre compétence et IPP, ainsi que les points de vue. Qu'une modification soit faite sur le point de vue consensus (ou modèle de l'apprenant principal) ou sur un de ses points de vue, elle sera gérée de la même façon.

Cette gestion de l'évolution du modèle dans le temps pourra permettre d'effectuer le suivi des évolutions des apprentissages. En effet, un modèle de l'apprenant contenant l'évolution du modèle cognitif de l'apprenant contient aussi les preuves des progrès de l'apprenant. Par exemple, en se centrant sur une compétence, il sera possible de montrer graphiquement l'évolution dans le temps du degré d'habileté et du niveau de performance que l'apprenant possède sur cette compétence. Notre modèle offre la possibilité de consulter ces informations mais nous ne travaillons pas sur l'analyse de cette évolution des apprentissages qui représente une recherche à part entière. La figure III-8 représente notre proposition de modèle de l'apprenant. En plus des éléments présents sur les figures présentées dans la section 3.1. de ce chapitre concernant la caractéristique cognitive de notre proposition, nous trouvons les éléments liés aux caractéristiques points de vue multiples et évolutives du modèle. Ainsi, le modèle contient un lien à des points de vue qui appartiennent à des acteurs ou à des groupes d'acteurs. Le modèle consensus et les points de vue ont

un lien sur eux-mêmes intitulé *a pour version antérieure* qui représente l'évolution du modèle et la prise en compte des différentes versions créées.

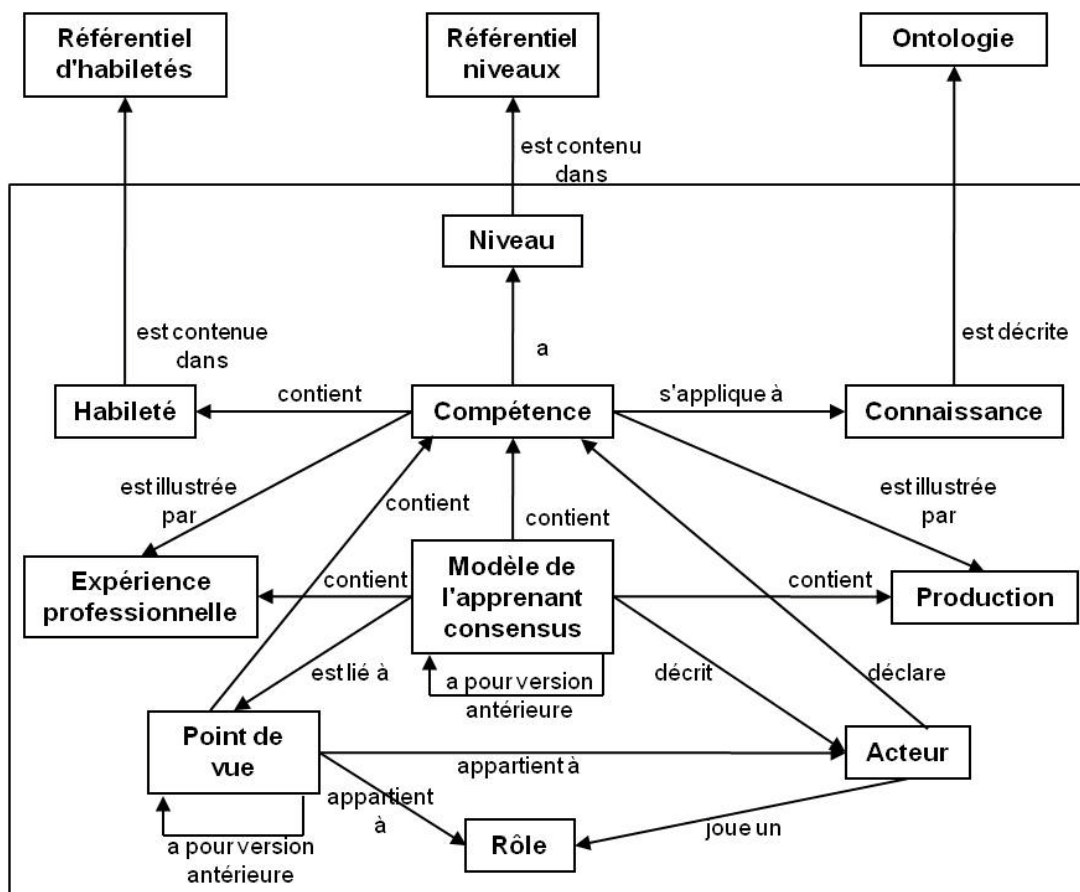


Figure III-8. Notre proposition de modèle de l'apprenant cognitif, à points de vue multiples et évolutif

3.2. Notre proposition : opérationnalisation

Nous avons présenté dans la section précédente notre proposition de modèle de l'apprenant au niveau conceptuel, ainsi que ses aspects points de vue multiples et évolution. Dans cette section, nous allons présenter comment opérationnaliser un tel modèle.

3.2.1. Métamodèle, modèle et instance : structure en 3 couches

Pour permettre l'utilisation du modèle de l'apprenant que nous venons de présenter dans différents contextes, nous avons développé une structure de modèle en trois couches : métamodèle, modèle et instance. Il est à noter que cette section est conceptuelle et vise à expliquer la dynamique d'utilisation de la structure présentée dans la section précédente. Le **métamodèle** de l'apprenant est une structure générique paramétrisable. Les modèles de l'apprenant, créés à partir de ce métamodèle, sont aussi des structures mais aux paramètres fixés selon un contexte d'utilisation particulier (une formation, un cours), ces modèles sont appelés **modèles de l'apprenant contexte**. Ces modèles contexte peuvent ensuite être instanciés pour des individus, on parle alors de **modèle de l'apprenant ou d'instance** (voir la figure III-9).

Ainsi, comme précisé dans la section 1.1. Les concepts de modèle et de métamodèle du chapitre I à la page 22, les modèles contexte sont des modèles *conformes au* métamodèle, plus précisément, un modèle contexte permet de décrire un cas particulier soit, dans notre recherche, un contexte d'apprentissage particulier. L'utilisation d'un métamodèle permet la création de modèles contexte pour un ensemble non fini de contextes particuliers. Ces différents modèles contexte peuvent être créés dans différentes institutions à partir du même métamodèle (soit à partir de la même structure) permettant ainsi d'être connectés ou regroupés dans un même modèle de l'apprenant, ainsi que d'être comparés. Dans l'état actuel de l'apprentissage qui se déroule dans des contextes de plus en plus variés et tout au long de la vie, il semble intéressant et même pertinent de proposer une première brique visant la création d'un modèle de l'apprenant « universel », dans le sens d'utilisable dans tous les contextes et toute la vie.

Le **métamodèle** est le modèle conceptuel que nous avons présenté au début de ce chapitre, il contient principalement les compétences, les productions et les IPP de l'apprenant ainsi que des liens entre ces éléments. L'utilisation du terme métamodèle

est d'ordre méthodologique et a pour but de conceptualiser notre proposition. Elle n'est absolument pas d'ordre technique. Nous allons décrire les deux couches restantes plus précisément dans les sections qui suivent.

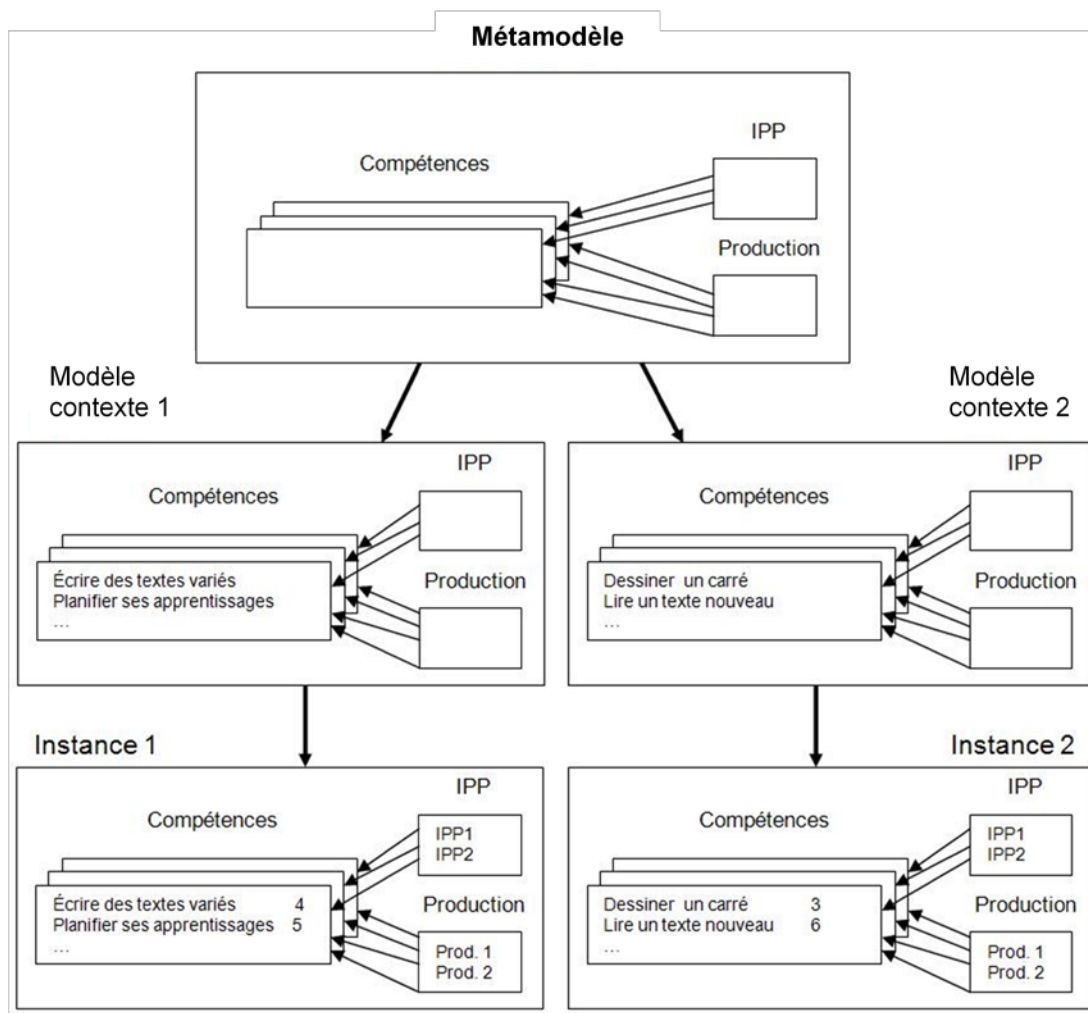


Figure III-9. Métamodèle, modèle et instance

3.2.2. Modèle contexte

Le modèle contexte reprend la structure du métamodèle et l'instancie pour un contexte particulier. Un modèle contexte contient ainsi une structure de compétences, soit un référentiel de compétences, qui seront déclarées pour chaque individu. Par exemple, si un modèle contexte est créé pour un cours particulier, il contiendra les

compétences visées par le cours (mais sans évaluation, uniquement des énoncés de compétence). Le modèle contexte contient en plus du référentiel de compétences, le référentiel d'habiletés et l'ontologie (ou les ontologies) auxquelles les compétences seront référencées. Lors de l'instanciation de ce modèle contexte pour un individu, chaque compétence pourra être évaluée. Le modèle contexte pourrait aussi être utilisé pour créer des instances abstraites, comme par exemple une instance seuil et une instance visée qui contiennent les évaluations aux compétences du cours requises pour la première, et visées pour la seconde. Ces instances abstraites permettraient de comparer les modèles d'individus à des modèles seuils ou visés. Par exemple, la comparaison d'un modèle seuil avec un modèle d'un apprenant particulier permettra de s'assurer que l'apprenant en question possède les compétences nécessaires pour s'inscrire à un cours. De même, la comparaison avec un modèle de compétences visées permettra de s'assurer que l'apprenant a réussi le cours (très simplement on pourrait considérer que si toutes les compétences du modèle visé se trouvent dans le modèle de l'apprenant alors celui-ci a réussi le cours, bien sûr d'autres modalités pourraient être appliquées). Nous n'avons pas approfondi cette possibilité dans la suite de notre recherche.

Les modèles contexte se distinguent selon différents paramètres (comme le nombre de points de vue, la stratégie de négociation adoptée...). Pour gérer ces paramètres, nous utilisons la notion de contrat. La section suivante explique les paramètres distinguant les modèles contexte et comment les contrats seront utilisés dans notre recherche.

3.2.2.1. Interactions par l'utilisation de contrats

Le modèle de l'apprenant sera en interaction avec différents acteurs et avec différents systèmes d'apprentissage en ligne. De plus, il appartient à l'apprenant, c'est-à-dire que l'acteur représenté par un modèle aura toujours son mot à dire concernant son modèle. Ainsi, les interactions du modèle avec différents acteurs (humains ou

machines) doivent être spécifiées et acceptées par le propriétaire du modèle (pas à chaque interaction mais uniquement à chaque première interaction avec un nouvel acteur). Pour cela, nous proposons d'utiliser la notion de contrat en informatique. En effet, ces contrats permettent de spécifier les interactions entre différents composants informatiques, c'est-à-dire les obligations de chaque intervenant. Par exemple, lorsqu'un apprenant veut suivre un cours dans une université, celle-ci peut exiger d'avoir accès à toutes les informations contenues dans le modèle de l'apprenant et approuvées par un professeur ou un tuteur, elle peut aussi exiger que les professeurs ou tuteurs qui enseigneront à l'apprenant et le soutiendront puissent ajouter des informations dans le modèle. Ainsi, si l'apprenant veut effectivement s'inscrire à un cours dans cette université, il est obligé d'accepter ces conditions. Le modèle pourra donc être relié au système d'apprentissage en ligne de l'université avec un contrat qui spécifiera ces conditions.

Nous avons identifié quatre éléments paramétrisables et spécifiables dans les contrats. Nous les appelons éléments paramétrisables car leur valeur peut varier selon chaque contexte d'utilisation du modèle de l'apprenant. Notre modèle a été conçu afin de pouvoir générer des modèles de l'apprenant de différents types qui ne sont pas prédéfinis à l'avance, c'est-à-dire qu'il est possible de combiner toutes les valeurs possibles des différents éléments paramétrisables. Des exemples de contrats sont proposés dans le chapitre V. Nous allons présenter les quatre éléments identifiés ainsi que les valeurs qu'ils peuvent prendre. Ces éléments sont les droits des acteurs, la granularité de l'évolution, le nombre de points de vue et la stratégie de négociation.

Le premier élément, *droits des acteurs*, fait référence à un ensemble de paramètres. Il permet de préciser quel type d'acteur a le droit de faire quel type d'action sur quel élément du modèle. Les interactions peuvent être de deux types : l'enrichissement du modèle ou alors sa consultation afin d'effectuer une autre tâche (pour personnaliser un cours par exemple). Les différents types d'acteurs peuvent être des professeurs, des tuteurs, des pairs apprenants, l'apprenant lui-même, des responsables de stage, des membres d'une équipe administrative ou des systèmes

d'apprentissage en ligne (acteur machine). Les actions qui pourront être effectuées sont la consultation, la modification et l'ajout d'informations. Ces informations sont les éléments du modèle, c'est-à-dire les compétences, les productions, les informations personnelles et professionnelles ainsi que les liens entre ces éléments. De plus, le contrat pourra spécifier à quel moment les modifications du modèle pourront avoir lieu. Le tableau suivant présente un exemple de droit d'action utilisé par un contrat.

Tableau III-1. Exemple de droits des acteurs pour un contrat particulier

Informations Acteurs	Compétences	Production	IPP	Liens
Apprenant	Consultation	Consultation Modification Ajout	Consultation Modification Ajout	Consultation Modification Ajout
Professeur	Consultation Modification Ajout	Consultation	Consultation	Consultation Modification Ajout
Tuteur	Consultation	Consultation	Consultation	Consultation
Pair	Consultation	Consultation	Consultation	Consultation
Système d'apprentissage en ligne	Consultation Modification Ajout	Consultation	Consultation	Consultation
Administration	Consultation	Consultation	Consultation	Consultation
Invité	Consultation	Consultation	Consultation	Consultation

Dans cet exemple, l'apprenant ne peut pas saisir de compétence dans son modèle, seuls le professeur et le système d'apprentissage en ligne peuvent effectuer cette action. Seuls le professeur et l'apprenant peuvent ajouter des liens et seul l'apprenant peut ajouter des productions et des IPP. Les autres types d'acteurs peuvent uniquement consulter le modèle. Il faut aussi préciser qu'il est possible de définir des droits particuliers pour un acteur particulier. Par exemple, si au cours d'un stage, le responsable du stage en entreprise est chargé d'évaluer certaines compétences de l'apprenant, il aura besoin de pouvoir entrer ces compétences dans le modèle alors que son rôle serait probablement *Invité*.

Le deuxième élément paramétrisable concerne le *niveau de granularité* auquel le système d'apprentissage en ligne peut faire évoluer le modèle, c'est-à-dire en créer

une nouvelle version à la suite d'une ou de plusieurs modifications effectuées (selon les droits définis dans le premier paramètre). Nous avons présenté les niveaux de granularité possibles dans la section précédente (à la fin d'un programme de formation, à la fin d'un scénario pédagogique (un cours), après une activité d'apprentissage (un module ou chapitre de cours), après l'utilisation d'une ressource pédagogique, voir figure III-7 à la page 80). Le niveau de granularité précisé dans un contrat correspond au plus petit niveau acceptable mais il n'est pas obligatoire, c'est-à-dire que si le niveau précisé est après une activité d'apprentissage, il n'est pas obligatoire d'entrer des informations à la fin de chaque activité d'apprentissage. Cela signifie plutôt qu'il est possible de le faire.

Le troisième élément paramétrisable est le *nombre de points de vue*. Nous avons déjà mentionné cet élément précédemment. Il consiste ici à préciser dans le contrat comment se fait la création des points de vue, c'est-à-dire qu'il faut préciser le nombre de points de vue et pour chaque point de vue à qui il appartient (cela peut être à un groupe d'acteurs identifiées par leur rôle (par exemple professeur) ou alors à un individu particulier).

Finalement, le quatrième élément paramétrisable est la *stratégie de négociation*. Cette stratégie permet de gérer les éventuels conflits existants entre différents points de vue pour obtenir un consensus qui sera conservé dans le point de vue consensus. Il s'agit ici de préciser la nature de cette stratégie pour le contrat en question. Une stratégie peut être automatique ou elle peut demander l'avis d'acteur humain. Un exemple de stratégie automatique consiste, par exemple, à faire une moyenne de toutes les évaluations d'une compétence et à appliquer ce chiffre au point de vue consensus. Une autre stratégie pourrait contenir un mécanisme d'obtention de consensus auprès des acteurs humains ayant évalué une même compétence différemment. Des stratégies de négociation ont été développées dans le cadre des simulations effectuées dans notre recherche. Ces stratégies sont présentées dans le chapitre IV.

3.2.3. Modèle de l'apprenant pour un individu particulier (instance)

Nous allons expliquer plus en détail les instances de modèle de l'apprenant qui représentent un individu particulier. Dans notre solution, un modèle de l'apprenant est composé d'un modèle principal et d'un ou plusieurs modèles contexte temporaires (voir la figure III-10).

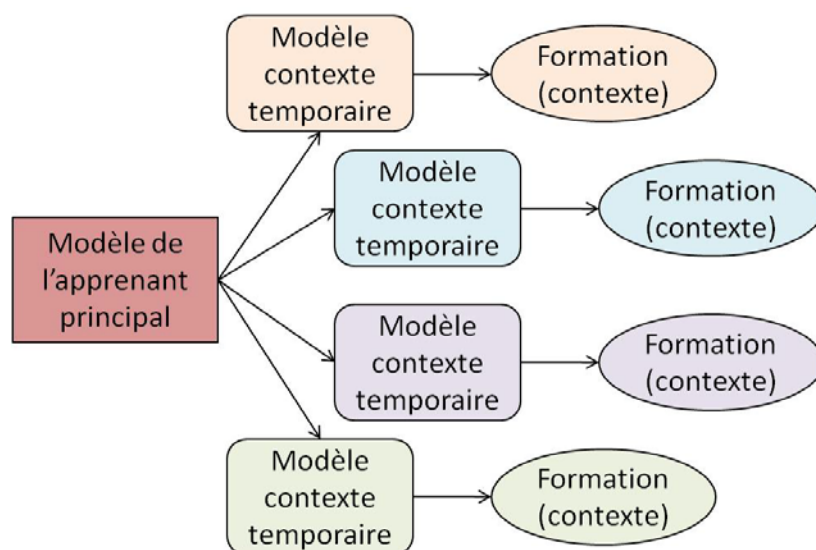


Figure III-10. Modèles de l'apprenant principal et contexte temporaires

Les modèles contexte temporaires sont liés à un contexte particulier, à une formation particulière (comme un cours), ce sont des modèles vivants qui évoluent en parallèle des apprentissages. Le modèle de l'apprenant principal est le résultat de ces évolutions. Ainsi nous disposons d'un modèle « mémoire », le modèle principal, et de modèles « courants », les modèles contexte temporaires. Cette façon de faire nous permet de gérer plus facilement les points de vue et la gestion de l'évolution dans le temps des modèles d'apprenant. Voyons plus précisément ce que sont ces deux types de modèles.

3.2.3.1. Le modèle de l'apprenant principal

Le modèle de l'apprenant principal est un modèle stable, fermé. Il n'évolue pas en parallèle des apprentissages, il n'est mis à jour qu'à la fin d'un programme de

formation (ou d'un cours) et non au fur et à mesure des apprentissages. Il est constitué de deux points de vue (celui de l'apprenant et celui des institutions). Nous expliquerons plus précisément comment ces deux points de vue évoluent dans une section suivante. Dans ce modèle, nous avons une gestion de l'évolution dans le temps, par contre, la gestion des points de vue est limitée puisque le modèle principal ne contient que deux points de vue.

La structure de ce modèle est similaire à celle du métamodèle présenté au début de ce chapitre. Le modèle principal est donc constitué de :

- compétences,
- productions,
- IPP,
- liens entre ces éléments,
- un point de vue de l'apprenant qui contient l'auto-évaluation des compétences ainsi que la date à laquelle celle-ci a eu lieu,
- et un point de vue des institutions qui contient les évaluations des compétences ainsi que leur date et contexte (quelle institution, quel acteur de cette institution).

Le modèle de l'apprenant principal est lié à un ou plusieurs modèles de l'apprenant contexte temporaires qui évoluent en même temps que les apprentissages et qui sont chacun liés à une formation particulière.

3.2.3.2. Le modèle de l'apprenant contexte temporaire

Au contraire du modèle principal, le modèle contexte temporaire est vivant et il évolue en parallèle des apprentissages. À chaque formation suivie par un apprenant, un modèle contexte temporaire est créé (à partir du modèle contexte lié à la formation en question et tel que présenté dans la structure en trois couches). Les modèles contexte temporaires ne sont pas persistants puisque lorsque la formation est terminée, le modèle temporaire disparaît. Ce type de modèle contient différents points de vue (autant que nécessaire). Il faut donc gérer les versions de points de vue des

modèles contexte temporaires. L'intégration du modèle contexte temporaire dans le modèle principal est décrite dans la section suivante. Le modèle temporaire a la même structure que le modèle principal en plus d'avoir différents points de vue.

3.2.4. Dynamique d'utilisation

Nous allons présenter les deux principaux processus d'utilisation des modèles de l'apprenant, principal et contexte temporaire. Tout d'abord, nous allons expliquer le processus opérant au démarrage d'un programme de formation. Ensuite, nous expliquerons celui opérant à la fin d'un programme de formation.

3.2.4.1. Débuter une formation

La première fois qu'un apprenant débute une formation pour laquelle notre modèle de l'apprenant est utilisé, une instance de modèle est créée pour cet individu. Le modèle de l'apprenant comprend alors uniquement un modèle principal qui contient les informations personnelles de l'apprenant (nom, adresse, courriel...) mais qui est vide pour le reste. Si un modèle pour l'apprenant existe déjà, celui-ci est récupéré dans un répertoire des modèles d'apprenants. Dans tous les cas, lorsqu'une formation est créée, un modèle contexte est créé avec le référentiel de compétences de la formation en question et le contrat-type associé. Ces éléments sont conservés ensemble pour pouvoir être récupéré lorsque nécessaire dans un répertoire de modèles contexte. Lorsqu'un apprenant débute la formation en question, le processus *Débuter une formation* est lancé (voir la figure III-11).

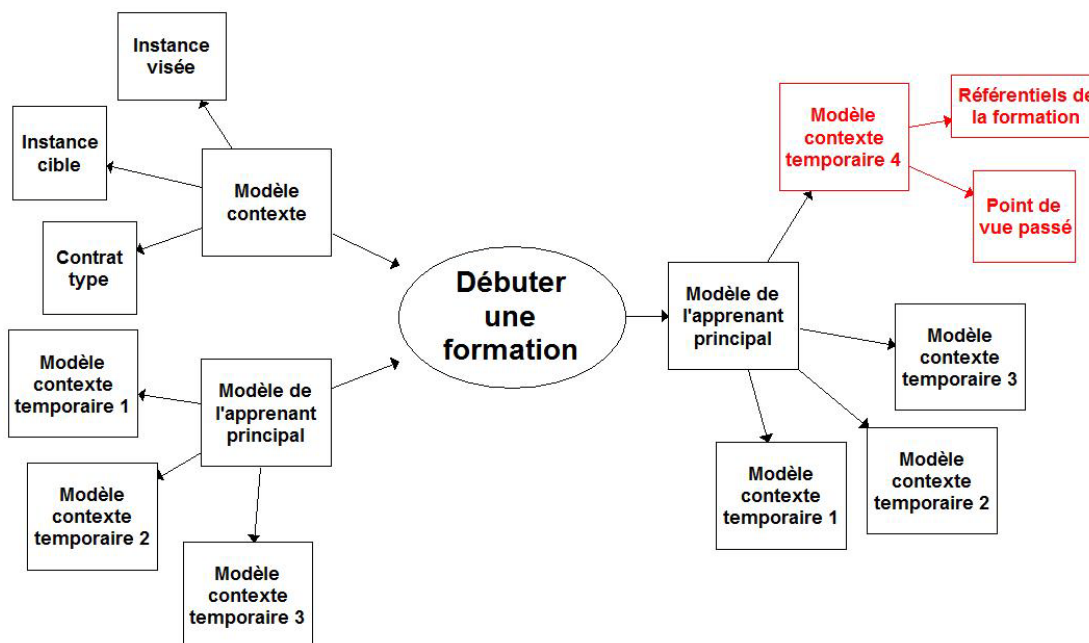


Figure III-11. Processus *Débuter une formation*

Se trouvent en entrée de ce processus, le modèle contexte, les instances abstraites qui y sont éventuellement associées et le contrat-type (récupéré dans le répertoire), ainsi que le modèle de l'apprenant principal et ses éventuels modèles contexte temporaires (récupérés dans le répertoire). Le modèle contexte permet d'instancier un modèle contexte temporaire lié à la formation débutée. Ce nouveau modèle contexte temporaire est lié au modèle de l'apprenant principal et il contient le contexte de la formation, c'est-à-dire les référentiels (de compétences, d'habiletés et les ontologies) utilisés par cette formation. De plus, le contrat type de la formation y est associé pour gérer les interactions de ce modèle contexte temporaire avec le système d'apprentissage en ligne et les acteurs humains. Aussi, un point de vue passé est créé dans le nouveau modèle contexte temporaire. Ce point de vue contient les évaluations des compétences du référentiel de la formation qui sont évaluées dans le modèle de l'apprenant principal. Cette information sera nécessaire au moment où la formation sera terminée, nous expliquerons plus en détail pourquoi dans la section suivante.

3.2.4.2. Terminer une formation

Lorsqu'un apprenant termine une formation, le modèle contexte temporaire correspondant est intégré dans le modèle principal. Pour cela, le processus *Terminer une formation* est lancé (voir la figure III-12).

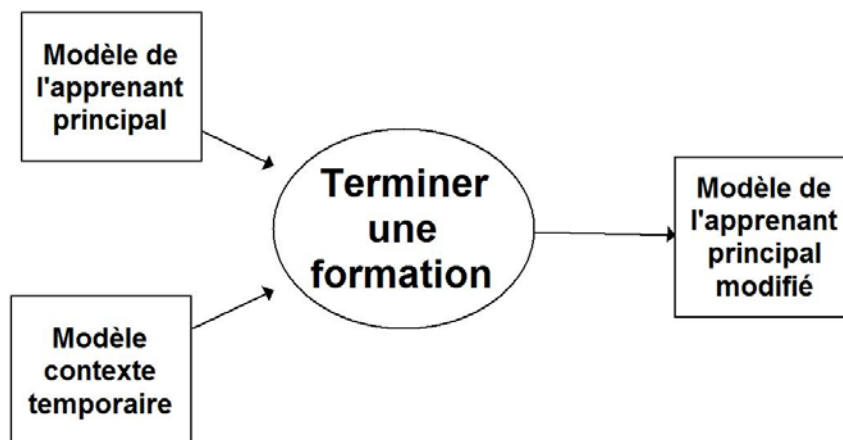


Figure III-12. Processus *Terminer une formation*

Le modèle de l'apprenant principal ainsi que le modèle contexte temporaire doivent être fournis au début de ce processus. En sortie, nous obtenons le modèle de l'apprenant principal dans lequel le modèle temporaire a été intégré. Le modèle contexte temporaire est alors détruit. La stratégie de négociation (issue du contrat) est incluse dans le modèle temporaire. Celle-ci est nécessaire à la fermeture du modèle temporaire. En effet, le modèle principal ne contient que deux points de vue, celui de l'apprenant et celui des institutions. Lors de l'intégration du modèle contexte temporaire au modèle principal, il faut décider quel point de vue doit être conservé (par exemple, celui des professeurs) pour les institutions. Ce choix se fait par l'application de la stratégie de négociation. Si le résultat de la fonction négociation est en conflit avec le point de vue passé qui contient l'évaluation de la même compétence effectuée précédemment, par défaut la meilleure évaluation est conservée mais l'apprenant peut toujours choisir quelle évaluation il souhaite garder. Pour le deuxième point de vue qui est celui de l'apprenant, toutes les compétences pour lesquelles l'apprenant s'est auto-évalué sont ajoutées à son point de vue dans le modèle principal.

3.2.5. Architecture globale

Dans cette section, nous donnons une vue d'ensemble de l'architecture globale de notre proposition (voir la figure III-13). Plus précisément, nous présentons les composantes principales du système, leurs relations et les services qu'elles fournissent afin de permettre la mise en opération du modèle conceptuel.

Le premier ensemble de modules représenté sur la figure regroupe des modules de création : création des modèles d'apprenant, des modèles contexte et des modèles contexte temporaires. Les modèles créés par ces différents modules sont stockés dans des répertoires. Les modèles contexte sont regroupés dans le répertoire des modèles contexte où ils peuvent être récupérés à chaque fois que le système a besoin de créer un nouveau modèle contexte temporaire. Les modèles d'apprenant principaux et contexte temporaires sont stockés dans un répertoire de modèles d'apprenant. Le module de création de modèles d'apprenant n'est utilisé qu'une seule fois par apprenant ayant un modèle, lors de la première utilisation du modèle. Au cours des utilisations futures, le modèle est directement récupéré dans le répertoire des modèles d'apprenant. De même, pour un contexte particulier, le module de création des modèles contexte ne sera utilisé qu'une seule fois. Par contre, le module de création des modèles contexte temporaires est utilisé à chaque fois qu'un apprenant débute une formation (faisant l'objet d'un modèle contexte). Pour fonctionner, ce module doit disposer du modèle du contexte et du modèle de l'apprenant auquel il ajoute un modèle contexte temporaire.

Une fois créés, les modèles contexte temporaires peuvent être consultés (tout comme les modèles d'apprenant principaux) ou modifiés, pour cela, ils sont récupérés dans le répertoire des modèles d'apprenant. Les modèles d'apprenant généraux sont aussi utilisés par le module de gestion des modifications qui permet de « stocker » leurs évolutions dans le temps, évolutions dans le sens de modifications apportées au modèle de l'apprenant au fur et à mesure de l'évolution de l'apprentissage.

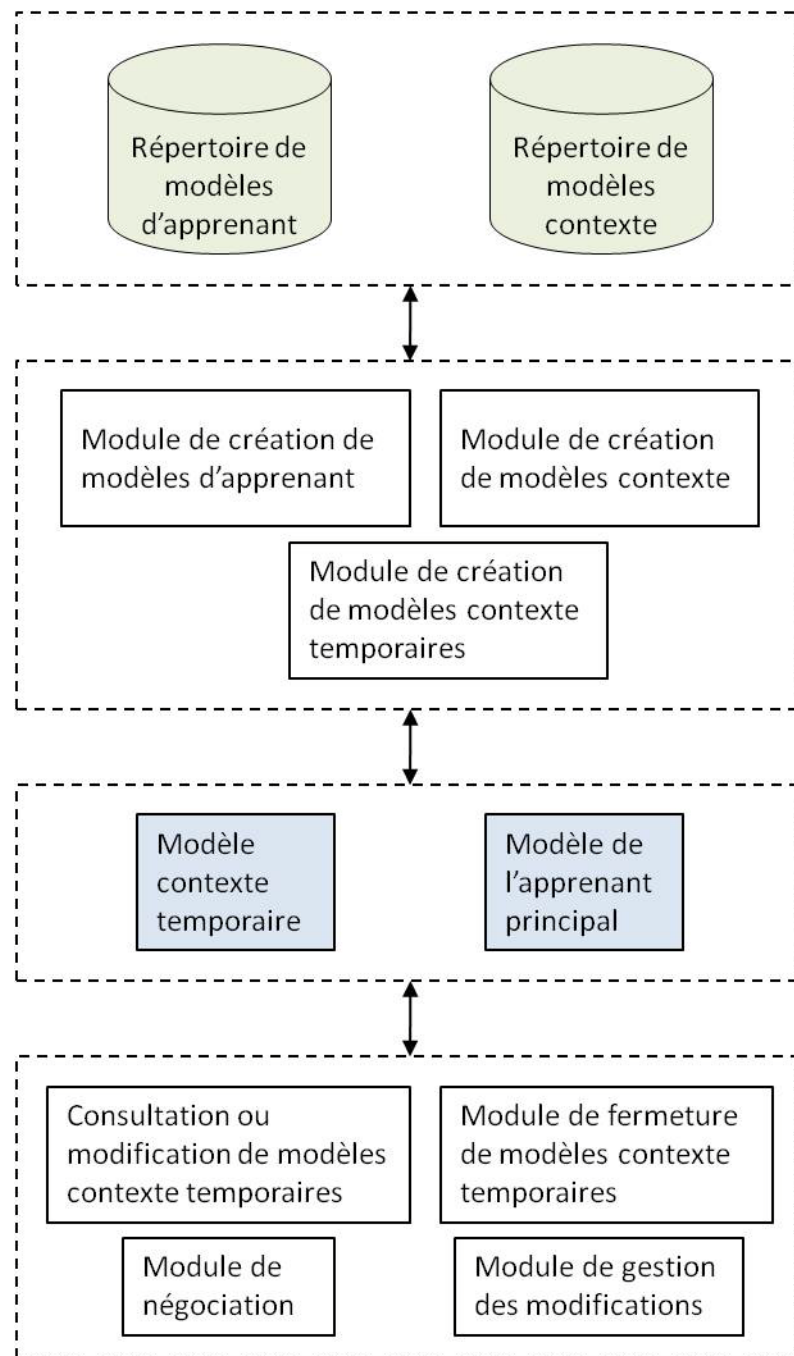


Figure III-13. Architecture globale

Le module de fermeture des modèles contexte temporaires fait appel au module de négociation et dépose le modèle d'apprenant principal (dans lequel le modèle contexte temporaire a été ajouté) dans le répertoire des modèles d'apprenant.

3.3. Conclusion

Dans ce chapitre, nous avons présenté la structure du modèle de l'apprenant que nous proposons et nous l'avons appelée métamodèle. Ce métamodèle permet de créer des modèles de l'apprenant pour différents contextes (ou formations) à partir desquels on peut créer des instances de modèle (c'est-à-dire des modèles pour des individus particuliers, que nous avons appelés modèles contexte temporaires). À la fin de la formation, le modèle contexte temporaire qui y est associé est intégré au modèle de l'apprenant principal. Notre proposition constitue un modèle cognitif de l'apprenant concerné. C'est aussi un modèle à points de vue multiples et évolutif. Pour permettre l'utilisation du modèle dans différents contextes, nous adoptons les concepts de modèle de l'apprenant contexte et de modèle de l'apprenant contexte temporaire. Les modèles contexte temporaires sont dynamiques, ils évoluent en parallèle des apprentissages et ils sont constitués de différents points de vue. Les modèles principaux contiennent les résultats de ces évolutions. Ils sont constitués de deux points de vue (celui de l'apprenant modélisé et celui des institutions) et une gestion de leur évolution est assurée (évolution dans le sens d'ensemble de modifications apportées au modèle en parallèle de l'évolution des apprentissages). La figure III-14 représente l'ensemble de ces notions. Nous avons aussi présenté une vue d'ensemble de l'architecture globale de notre proposition.

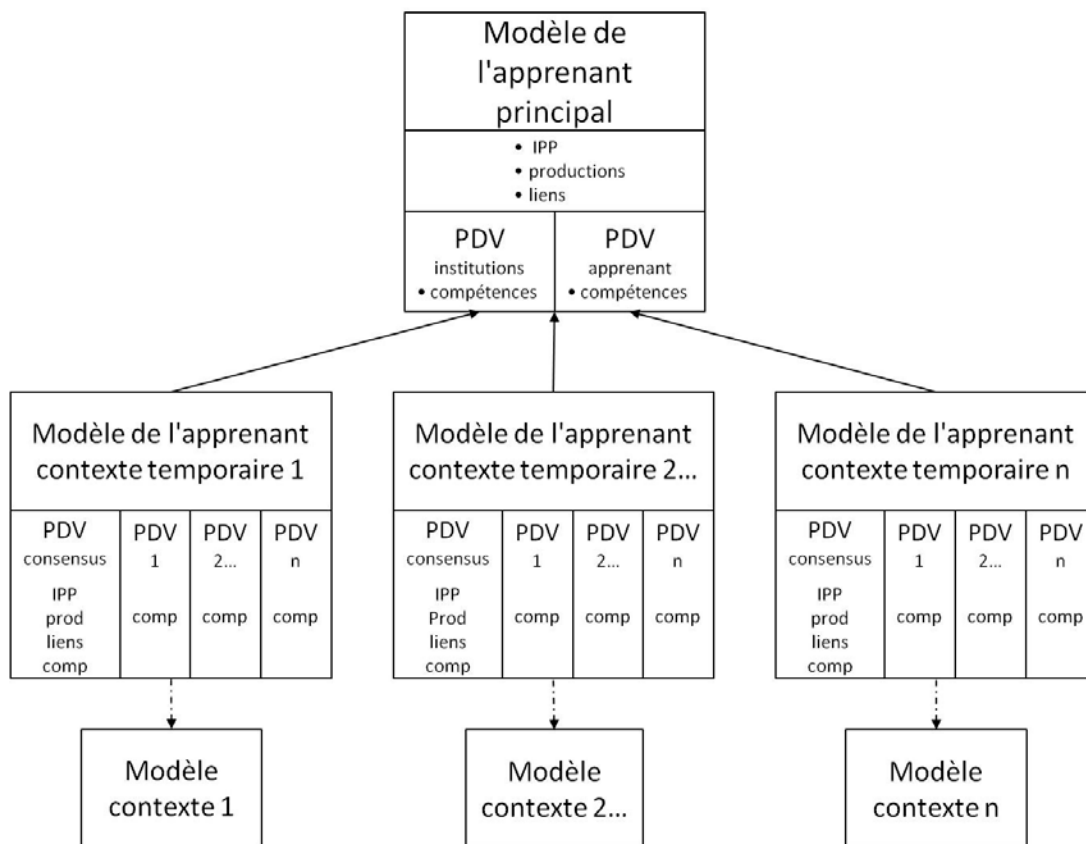


Figure III-14. Les modèles et leurs interactions

Ce chapitre s'est concentré sur le modèle conceptuel de notre proposition de modèle de l'apprenant, les chapitres suivants concernent sa mise en opération. Plus précisément, le chapitre IV présente le prototype informatique mettant en œuvre notre modèle conceptuel et le chapitre V présente l'évaluation du modèle conceptuel effectuée par simulations sur le prototype.

Chapitre IV - IMPLEMENTATION D'UN PROTOTYPE : NOSMA

Nous présentons dans ce chapitre Nosma, notre prototype de système de modélisation de l'apprenant. Pour commencer, nous affinons la présentation de la méthode appliquée et présentée dans l'introduction (section 4.1). Plus précisément, nous expliquons quelle méthode nous avons utilisée pour mener à bien la deuxième phase, soit celle d'élaboration et de construction dont nous présentons les résultats dans ce chapitre. Ensuite, nous présentons comment nous avons conçu Nosma (4.2) puis comment nous avons implémenté ce prototype (4.3).

4.1. Méthode

Nous avons présenté dans l'introduction la méthode de recherche utilisée dans le cadre de ce projet (voir la figure page 17). Ce chapitre présente les résultats de la deuxième phase, celle d'élaboration et de construction. Cette phase est itérative et produit, à chaque itération, un sous-système complet. Nous avons développé le système Nosma en quatre itérations. Pour chaque itération, les phases d'analyse, de conception, d'implémentation et d'intégration ont été effectuées. De plus, chaque itération se termine par le développement d'un système fonctionnel mais partiel. C'est à la fin de la dernière itération que nous obtenons un système fonctionnel et complet. La première itération a consisté à développer le squelette du modèle de l'apprenant, soit un modèle comprenant des compétences, des productions, des informations personnelles et professionnelles ainsi que des points de vue. La seconde

itération a traité le passage des objets Java au XML. Plus précisément, nous avons ajouté au système existant un module permettant de transformer du XML en Java et vice versa. La troisième itération a permis d'ajouter au prototype un système de gestion de versions afin de répondre à la caractéristique d'évolution dans le temps du modèle. Finalement, la quatrième et dernière itération a consisté à affiner le modèle selon deux scénarios d'utilisation. Ces scénarios sont réutilisés pour effectuer des simulations (voir chapitre V). Nous présentons dans ce chapitre les résultats de ces quatre itérations confondus et non les résultats intermédiaires obtenus entre chaque itération.

La première étape de cette phase d'élaboration et construction est l'analyse. Nous avons développé des cas d'utilisation puis nous avons analysé différents outils pertinents pour notre projet. Les cas d'utilisation sont présentés à la section 4.2.2. Les cas d'utilisation. Une partie de l'analyse des outils a été présentée dans le chapitre II (concernant les *ePortfolios* et les systèmes de gestion de versions), nous présentons dans ce chapitre un nouvel outil que nous avons utilisé dans le développement de Nosma à la section 4.3.1. Outils utilisés. Une partie des résultats de cette étape d'analyse a déjà été présentée dans le chapitre précédent, soit le modèle conceptuel de notre modèle de l'apprenant. L'étape de conception a suivi cette première étape d'analyse. Nous y avons développé un modèle des classes de conception (section 4.2.3.) que nous avons ensuite implémenté en Java puis testé dans les étapes suivantes, celles d'implémentation et d'intégration.

4.2. Conception de Nosma

Nous avons présenté dans le chapitre précédent le modèle conceptuel de notre proposition de modèle de l'apprenant. Nous poursuivons dans cette section la présentation des spécifications conceptuelles du système que nous proposons, Nosma. Nous allons, plus particulièrement, présenter les acteurs susceptibles d'utiliser le

modèle de l'apprenant, les cas d'utilisation du système ainsi que le diagramme de classes. Ensuite, nous présentons trois stratégies de négociation que nous avons conçues et implémentées dans Nosma. Pour finir, nous présentons des diagrammes d'interaction du système (plus précisément, des diagrammes de séquence).

4.2.1. Les acteurs

Nous avons identifié cinq types de rôle (acteurs humains ou machines) pouvant interagir avec Nosma :

- **Apprenant** : acteur à qui appartient le modèle.
- **Professeur / tuteur** : qui enseigne présentement à l'apprenant.
- **Pair** : autre apprenant qui est engagé présentement dans une activité d'apprentissage avec l'apprenant.
- **Invité** : toute personne invitée, par exemple un professeur / tuteur n'enseignant pas actuellement à l'apprenant, un pair n'étant pas dans un des groupes d'étudiants de l'apprenant, un responsable de stage issu de l'industrie, un membre de la famille de l'apprenant...
- **Système d'apprentissage en ligne (SAL)** : acteur machine.

Il est important de noter qu'un acteur peut jouer différents rôles.

4.2.2. Les cas d'utilisation

Nous présentons ici les cas d'utilisation de notre modèle de l'apprenant. Nous avons identifié cinq cas d'utilisation principaux (voir figure ci-dessous) :

1. Créer un modèle de l'apprenant
2. Créer un modèle contexte
3. Créer un modèle contexte temporaire
4. Consulter le modèle
5. Modifier le modèle

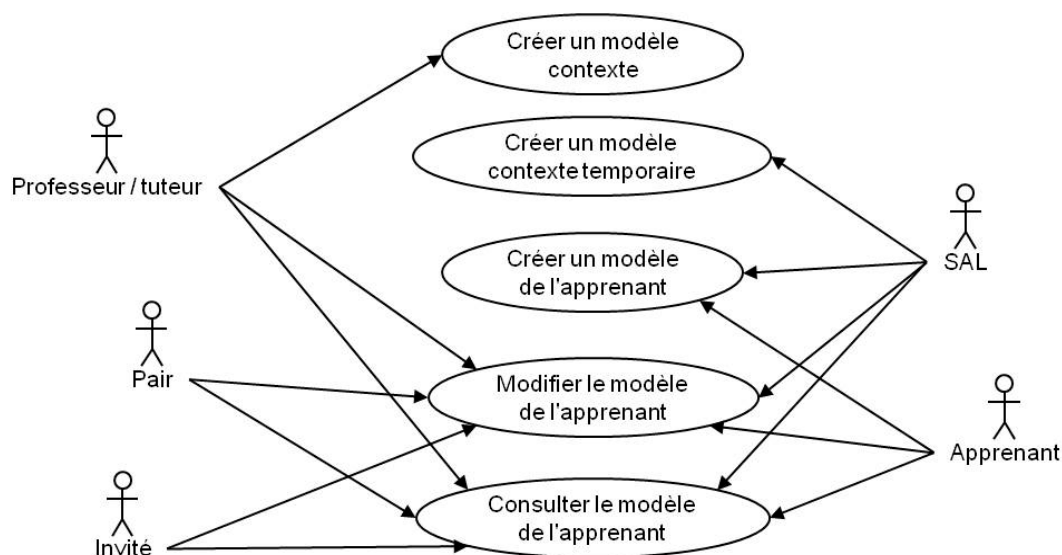


Figure IV-1. Diagramme des cas d'utilisation principaux

Les cas d'utilisation 2 et 3, *Créer un modèle contexte* et *Créer un modèle contexte temporaire*, n'ont pas été développés. Dans la version du prototype que nous présentons, les modèles contexte temporaires ne sont pas différenciés du modèle de l'apprenant principal. Ces modèles sont différenciés conceptuellement (comme présenté dans le chapitre III) mais pas techniquement. Plus précisément, ils sont contenus dans la même structure. En fait, au cours de l'utilisation d'un modèle contexte temporaire, tous les points de vue des acteurs sont créés et le point de vue consensus est contenu dans le modèle principal et n'est pas un point de vue séparé (même si conceptuellement, ils sont séparés). À la fermeture du modèle contexte temporaire, il n'y a pas à transférer le point de vue consensus dans le modèle principal, puisqu'il y est déjà. Nous n'avons pas implémenté cela dans notre prototype parce que ce n'était pas absolument essentiel à notre projet pour démontrer la faisabilité du modèle de l'apprenant que nous proposons selon nos objectifs (nous expliquerons plus clairement dans la suite du document comment et si nos objectifs de recherche ont été atteints). C'est pourquoi les cas d'utilisation 2 et 3 n'ont pas été développés dans cette version du prototype mais ce serait un ajout à apporter à notre système. De plus, le prototype que nous développons pour illustrer notre modèle

conceptuel inclut le modèle cognitif, le modèle à points de vue multiples, le modèle évolutif mais pas les droits d'accès des acteurs ni les contrats (ces deux aspects sont décrits au niveau conceptuel et sont utilisés dans les simulations effectuées avec le prototype, mais ils ne sont pas implémentés dans le prototype).

Le cas d'utilisation *Créer un modèle de l'apprenant* consiste à instancier un modèle pour un apprenant particulier et se fait une seule fois pour chaque apprenant. Lors de la création du modèle de l'apprenant, des informations concernant l'apprenant en question sont ajoutées au modèle, au minimum ses nom et prénom. L'apprenant lui-même et un système d'apprentissage en ligne peuvent effectuer ce cas d'utilisation. Il est important de noter que le cas d'utilisation *Créer un modèle de l'apprenant* précède les cas d'utilisation *Modifier* et *Consulter le modèle*. En effet, un modèle de l'apprenant doit avoir été créé avant de pouvoir être manipulé (que ce soit pour le consulter ou le modifier).

Le cas d'utilisation *Consulter un modèle de l'apprenant* consiste à consulter, c'est-à-dire visualiser les éléments du modèle; soit les compétences, les IPP, les productions et les liens entre ces éléments (voir la figure IV-2).

La consultation peut se faire sur le modèle principal ou sur un point de vue. *Consulter les compétences* correspond à visualiser les compétences présentes dans le modèle principal et dans les points de vue. Ceci inclut la consultation des liens de la compétence en question avec des productions ou des IPP. *Vérifier une compétence* est un cas particulier de *Consulter les compétences* et consiste à questionner le modèle de l'apprenant au sujet d'une compétence particulière. Ce cas d'utilisation consiste à demander si une compétence spécifique est présente dans le modèle et si oui quelle est son évaluation. Cette vérification peut servir à adapter un scénario pédagogique. En effet, si l'apprenant ne possède pas telle compétence, on pourrait, par exemple, lui proposer une mise à niveau avant de poursuivre le cours. *Consulter les IPP* correspond à la consultation des informations personnelles et professionnelles contenues dans le modèle principal et inclut la consultation des liens entre les IPP et les compétences. *Consulter les productions* correspond à la consultation des

productions présentes dans le modèle principal et des éventuels liens entre ces productions et les compétences.

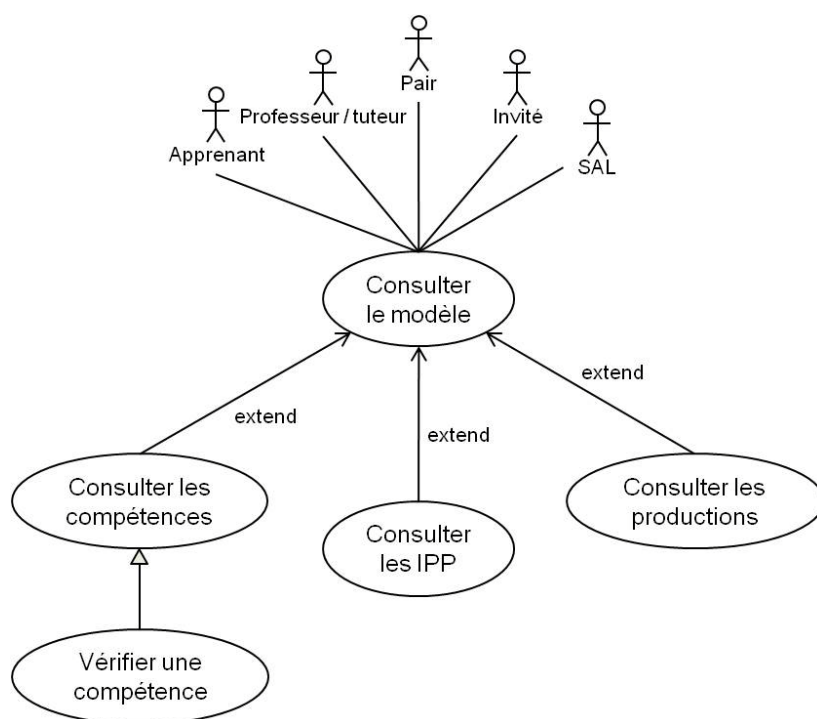


Figure IV-2. Cas d'utilisation Consulter le modèle de l'apprenant

Le cas d'utilisation *Modifier un modèle de l'apprenant* correspond à l'ajout ou à la mise à jour d'éléments dans le modèle de l'apprenant (voir la figure IV-3). Nous expliquerons ces cas d'utilisation plus en détail après avoir présenté le cas d'utilisation *Fermer le modèle contexte temporaire* qui est une spécialisation de *Modifier un modèle de l'apprenant*. *Fermer le modèle contexte temporaire* intervient lorsqu'un contexte est terminé, c'est-à-dire lorsqu'un modèle temporaire ne sera plus utilisé. Dans ce cas, comme expliqué dans le chapitre III, le modèle contexte temporaire est ajouté dans le modèle de l'apprenant principal. Cet ajout inclut la négociation des compétences présentes dans le modèle temporaire. Cette négociation a pour but de gérer les conflits pouvant intervenir si une même compétence est évaluée par différents acteurs et apparaît donc dans différents points de vue. Nous exposerons des stratégies de négociation dans un paragraphe suivant.

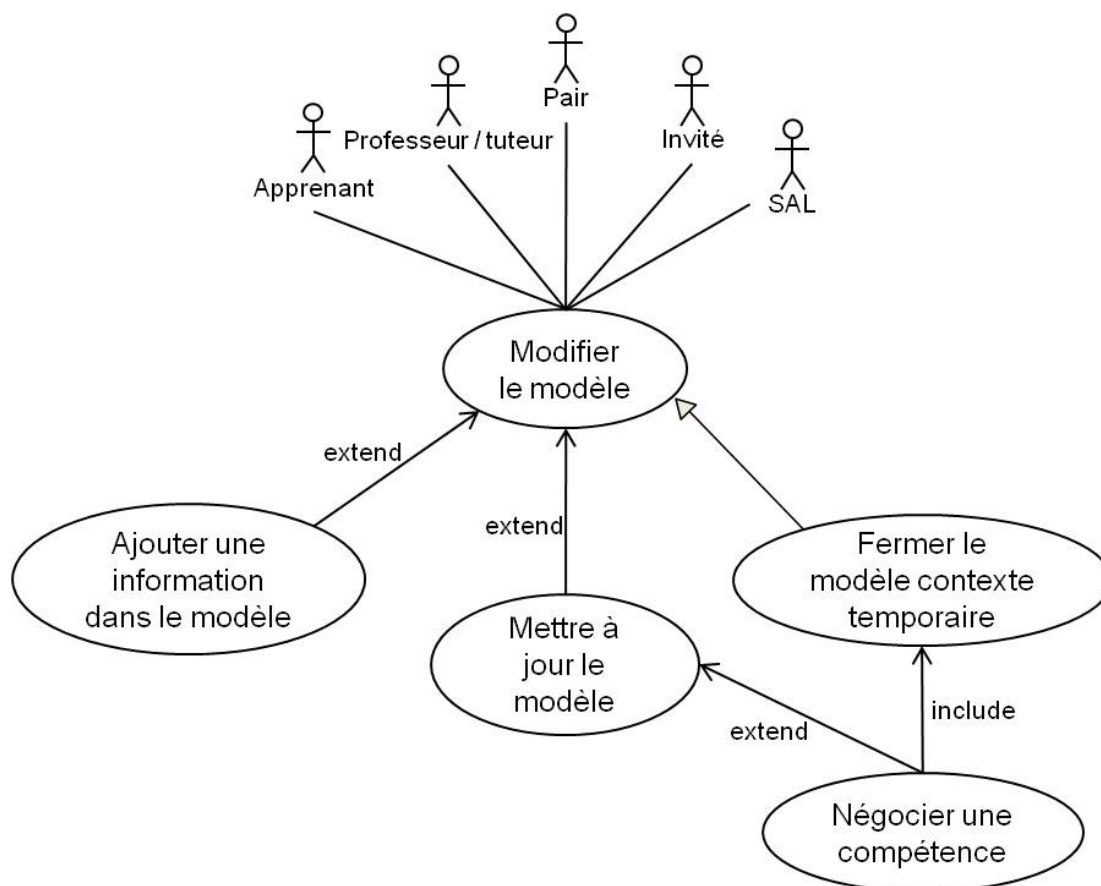


Figure IV-3. Cas d'utilisation Modifier le modèle de l'apprenant

Le cas d'utilisation *Négocier une compétence* fait appel à la stratégie de négociation du contexte concerné et s'applique à une compétence particulière. Notons toutefois qu'une compétence peut être négociée à tout moment et pas uniquement à la fermeture du modèle contexte temporaire. Dans ce cas, la négociation de la compétence correspond à une mise à jour du modèle. À la fermeture du modèle temporaire, seront négociées toutes les compétences non encore négociées et donc encore présentes dans des points de vue.

La figure IV-4 illustre le cas d'utilisation *Ajouter une information dans le modèle de l'apprenant*. Ce cas d'utilisation consiste à ajouter une compétence, une IPP, une production ou un lien (lien compétence-IPP ou lien compétence-production) dans le modèle.

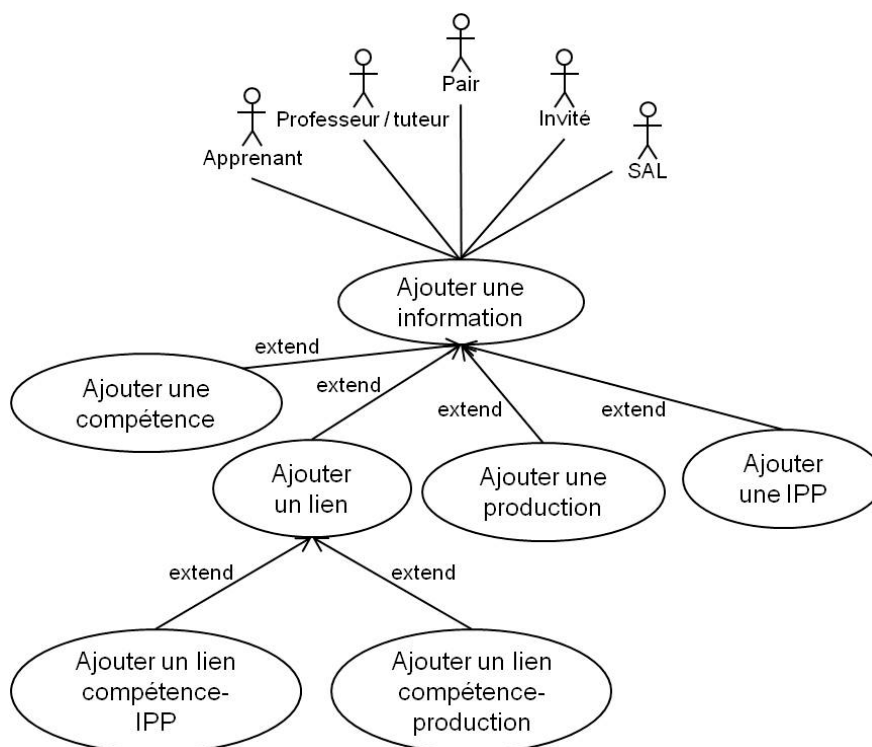


Figure IV-4. Cas d'utilisation Ajouter une information dans le modèle de l'apprenant

Ajouter une compétence correspond à l'évaluation d'une compétence, c'est-à-dire à préciser le niveau de la compétence soit le degré auquel la compétence est atteinte par l'apprenant concerné. Une compétence est toujours ajoutée dans le point de vue de l'acteur. Elle est placée dans le point de vue consensus après avoir été négociée.

Ajouter un lien, que ce soit un lien entre une compétence et une IPP ou entre une compétence et une production, correspond à illustrer une compétence. Cette illustration consiste à donner un exemple concret de la compétence pour montrer que celle-ci est effective, cet exemple (ou preuve) peut être une IPP ou une production.

Le dernier cas d'utilisation, *Mettre à jour une information*, correspond à la modification d'éléments déjà présents dans le modèle, soit les compétences, les IPP, les productions et les liens (voir la figure IV-5). Lors de l'utilisation du modèle, nous pouvons imaginer que tous les acteurs ne pourront pas modifier toutes les informations, cela sera spécifié dans les contrats. Par exemple, probablement que

seulement l'acteur dont c'est le point de vue sera en mesure de modifier les informations d'un point de vue particulier. De plus, la modification d'une compétence peut correspondre à sa négociation.

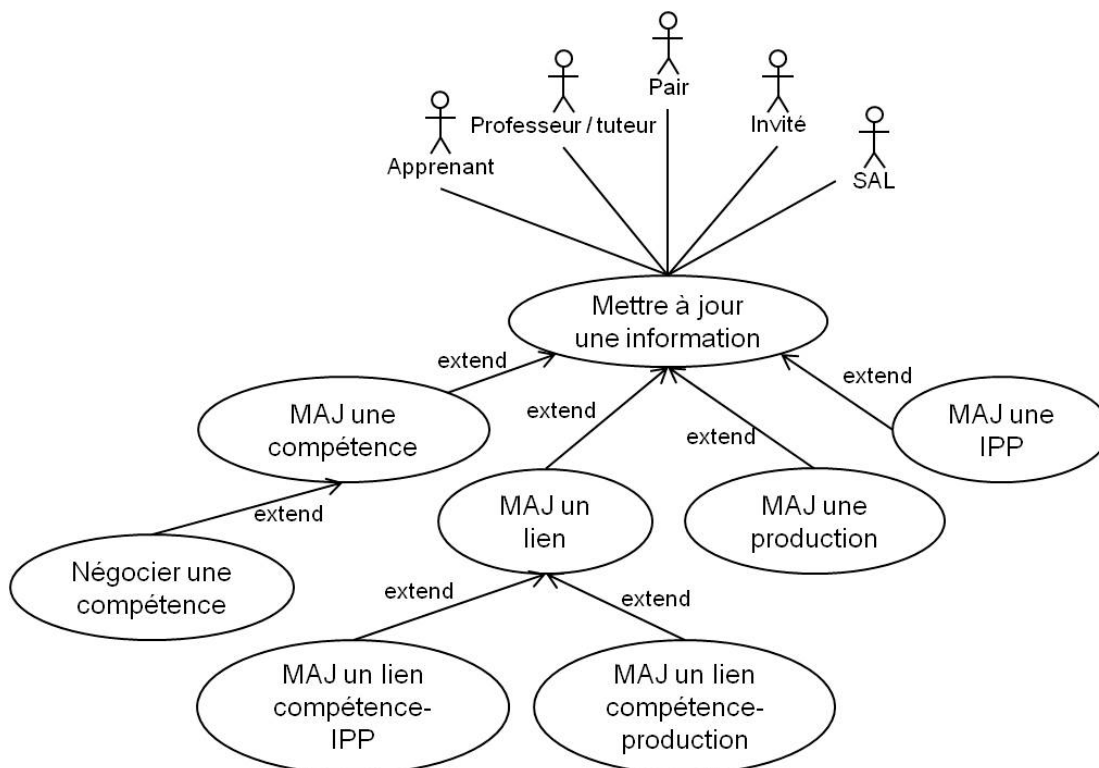


Figure IV-5. Cas d'utilisation Mettre à jour une information du modèle de l'apprenant

N'oublions pas, comme mentionné dans le chapitre III, que les droits des acteurs à effectuer différentes opérations sur les éléments du modèle sont spécifiés dans les contrats. À la base, tous les acteurs peuvent effectuer toutes les opérations (de consultation et de modification) sur les éléments du modèle. Ce sont les contrats qui permettent de restreindre les droits des acteurs.

4.2.3. Le diagramme de classes

Nous présentons dans cette section le diagramme de classes de Nosma. Pour des raisons de lisibilité, nous avons créé deux figures. La première contient toutes les classes mais pas tous les liens entre ces classes et la seconde présente les liens

manquants. Les deux permettent aussi de voir les attributs de chacune des classes, tels que précisés dans le chapitre III lors de la présentation du modèle conceptuel.

Le diagramme de classe principal est centré sur la classe *Modele* puisque c'est la classe centrale de Nosma (voir la figure IV-6).

Nous observons sur cette figure (page suivante) qu'un modèle est lié à un acteur, l'acteur représenté par le modèle en question. De même, un acteur ne possède qu'un seul modèle. La classe *Modele* est liée aux classes *Production*, *ExpProf* (expérience professionnelle), *LienProd* (lien entre une compétence et une production), *LienExpProf* (lien entre une compétence et une expérience professionnelle), *Competence* et *PointDeVue*, car le modèle contient des listes de chacun de ces éléments. La classe *Competence* est liée aux classes *Habilete* et *Niveau* puisqu'une compétence est décrite par une habileté et par un niveau. Le lien entre les classes *Competence* et *Acteur* représente le fait qu'un acteur est l'auteur d'une compétence, dans le sens que c'est lui qui a évalué la compétence. La figure montre aussi qu'un point de vue contient des compétences ainsi que des liens entre compétences et productions et entre compétences et expériences professionnelles. Finalement, la figure permet de voir que la classe *LienProd* représente le lien existant entre une compétence et une production et que la classe *LienExpProf* représente le lien existant entre une compétence et une expérience professionnelle.

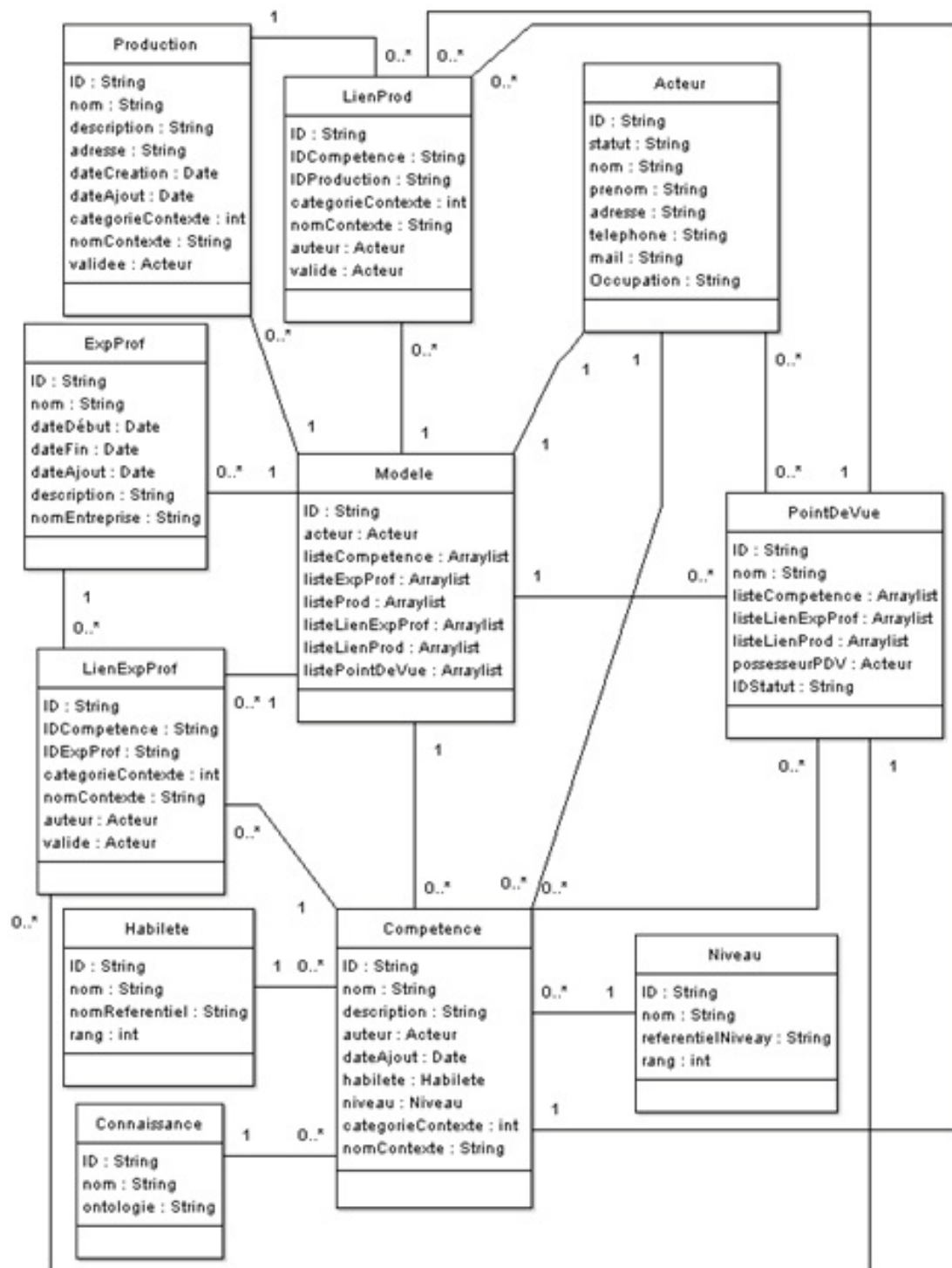


Figure IV-6. Diagramme de classes

4.2.4. Les stratégies de négociation conçues et implémentées

Nous avons conçu et implémenté trois stratégies de négociation qui seront utilisées dans les scénarios des simulations (voir chapitre V). Rappelons que la stratégie de négociation permet de placer une compétence dans le modèle principal qu'elle ait été évaluée par un ou plusieurs acteurs. Bien sûr, la négociation n'est nécessaire que si plusieurs acteurs ont évalué la compétence, mais tous les cas de figure doivent être couverts par une stratégie.

La première stratégie s'applique dans le cas d'un contexte académique où différents types d'acteurs peuvent évaluer une compétence, des professeurs, des tuteurs et l'apprenant dont c'est le modèle. Cette stratégie s'applique de façon automatique, c'est-à-dire qu'aucun acteur humain n'a besoin d'intervenir. Dans cette stratégie, l'évaluation de l'apprenant est toujours conservée dans son point de vue puisque le modèle principal comporte toujours le point de vue de l'apprenant concerné. Ensuite, les évaluations des professeurs sont priorisées par rapport à celle des tuteurs. Nous présentons les cas de figure existants et la solution adoptée pour chacun d'eux :

1. un seul acteur évalue la compétence (soit un seul professeur ou un seul tuteur) : dans ce cas la compétence est placée dans le modèle principal et supprimée du point de vue de l'acteur qui l'a évaluée.
2. deux acteurs ou plus de même statut évaluent la compétence (soit plusieurs professeurs **ou** plusieurs tuteurs) : dans ces cas de figure, on fait la moyenne des différentes évaluations. Faire une moyenne est questionnable car peut-on faire la moyenne de deux niveaux d'une compétence ? Un spécialiste en éducation serait nécessaire pour étudier cette question. Dans le cadre de notre projet et pour illustrer notre modèle conceptuel, nous considérons que cette pratique est acceptable. L'auteur conservé pour la compétence est un des acteurs ayant évalué la compétence (sélectionné aléatoirement par le système) mais tous les noms des acteurs ayant participé à la négociation (c'est-à-dire ayant évalué la compétence) sont conservés dans le contexte. En effet, lorsqu'il y a négociation faisant intervenir l'évaluation de plus d'un acteur, les noms de chaque acteur sont ajoutés au nom du contexte. Par exemple, si le nom du contexte est *cours de math* et que deux acteurs évaluent la compétence, *Dupuis* et *Prairie*, et que lors de la négociation, on

fait la moyenne des deux évaluations, alors le contexte sera *cours de math Dupuis Prairie*. On saura alors que l'évaluation associée à la compétence est issue d'une négociation entre les évaluations des acteurs s'appelant *Dupuis* et *Prairie*. De plus, la compétence est retirée des points de vue des acteurs qui l'avaient évaluée.

3. un ou des acteurs du type professeur et tuteur évaluent la compétence (soit un professeur ou plus **et** un tuteur ou plus) : dans ce cas, ce sont les évaluations du ou des professeurs qui priment, c'est-à-dire que dans tous ces cas de figure, c'est l'évaluation du ou des professeurs qui est conservée. Si un seul professeur a évalué la compétence, son évaluation est placée dans le modèle principal (cas 1) et supprimée de son point de vue et des points de vue du ou des tuteurs qui ont évalué la compétence. Si plusieurs professeurs ont évalué la compétence, une moyenne est faite, comme expliqué dans le cas précédent (cas 2). La compétence est alors supprimée des points de vue de tous les professeurs et tuteurs qui l'avaient évaluée.
4. l'apprenant dont c'est le modèle et un autre type d'acteur évaluent la compétence (soit l'apprenant **et** au choix, un ou plusieurs professeurs **ou** un ou plusieurs tuteurs mais il ne peut pas y avoir des professeurs et des tuteurs) : dans ces cas, l'évaluation faite par l'apprenant est toujours conservée et placée dans le point de vue de l'apprenant du modèle principal. Pour le reste, on rejoint le cas 2 et une moyenne est effectuée.
5. les trois types d'acteurs évaluent la compétence (soit l'apprenant **et** un ou plusieurs professeurs **et** un ou plusieurs tuteurs) : ici aussi l'évaluation de l'apprenant est conservée et pour le reste, on rejoint le cas 3, c'est-à-dire que ce sont les évaluations des professeurs qui sont conservées.

La seconde stratégie a été conçue pour être utilisée dans des contextes où c'est l'apprenant qui prend la décision finale, comme par exemple dans le cas où l'apprenant souhaite faire un bilan de compétence. Deux cas de figure peuvent se présenter, la compétence est évaluée par un seul acteur ou bien par plusieurs. Si la compétence n'est évaluée que par un seul acteur, cette évaluation est conservée dans le modèle principal. Mais dès que plusieurs acteurs interviennent, l'apprenant a toujours le choix. Lorsque la stratégie de négociation est appliquée à une compétence, toutes les évaluations dont elle a fait l'objet sont présentées à l'apprenant qui choisit celle qu'il souhaite conserver. Le système affiche donc des informations à l'utilisateur et attend une réponse de celui-ci avant de continuer. Une fois le choix

effectué, l'évaluation choisie est mise dans le modèle principal et la compétence est supprimée des points de vue de tous les acteurs qui l'avaient évaluée.

La troisième et dernière stratégie conçue concerne les contextes où l'évaluation est réalisée automatiquement par un système d'apprentissage en ligne. Dans ces contextes-ci, il n'y a pas de point de vue et il n'y a pas négociation car un seul acteur évalue les compétences, le système d'apprentissage. Ce sont donc les évaluations du système qui se retrouvent dans le modèle principal.

4.2.5. Les diagrammes de séquence

Nous allons présenter dans cette section les diagrammes de séquence de quelques scénarios d'utilisation de notre système. Le premier scénario que nous présentons est *Ajouter une production*. Au cours de ce scénario, un apprenant ajoute une production dans son modèle. La figure IV-7 présente le diagramme de séquence de ce scénario.

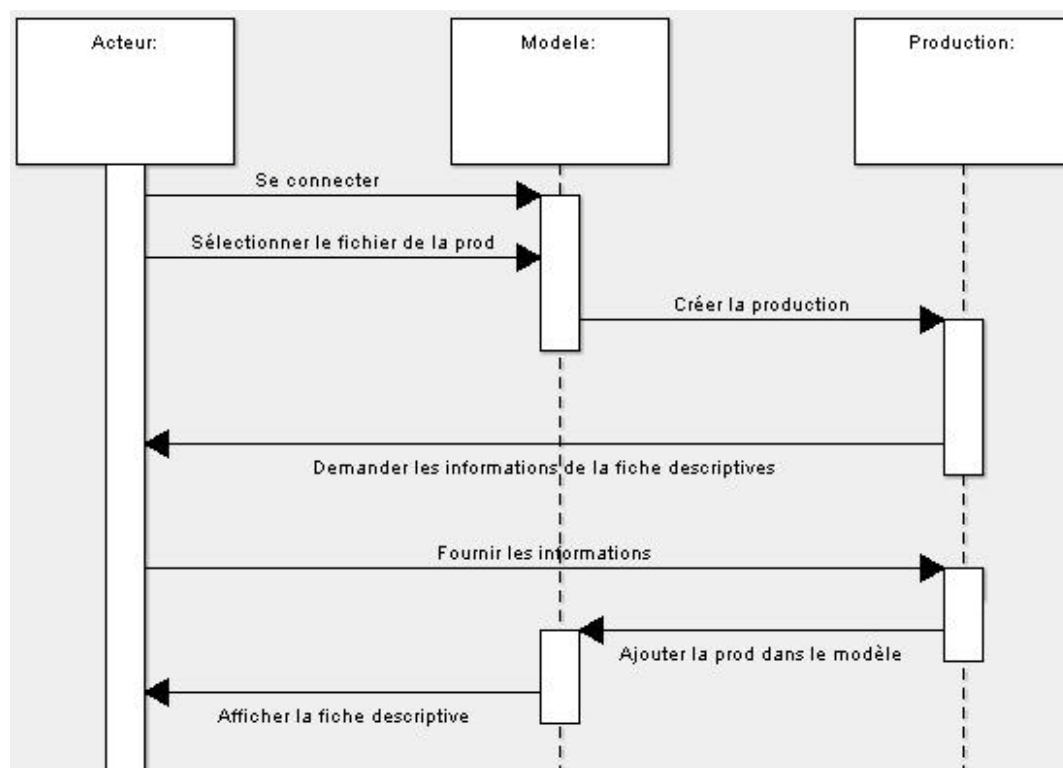


Figure IV-7. Diagramme de séquence Ajouter une production

Le scénario *Ajouter une expérience professionnelle* est similaire à celui-ci.

Après que l'apprenant se soit connecté au modèle de l'apprenant auquel il veut ajouter une production (soit son propre modèle), il sélectionne le fichier correspondant à la production. Une production est ainsi créée dans le modèle avec une fiche descriptive vide. Le système demande alors à l'utilisateur de préciser les informations de la fiche descriptive. Lorsque cela est fait, la production est ajoutée dans le modèle et la fiche descriptive est affichée à l'écran.

Le second scénario, *Ajouter/évaluer une compétence*, peut être effectué par tous les types d'acteur (apprenant, professeur, tuteur...). La figure IV-8 présente le diagramme de séquence du scénario.

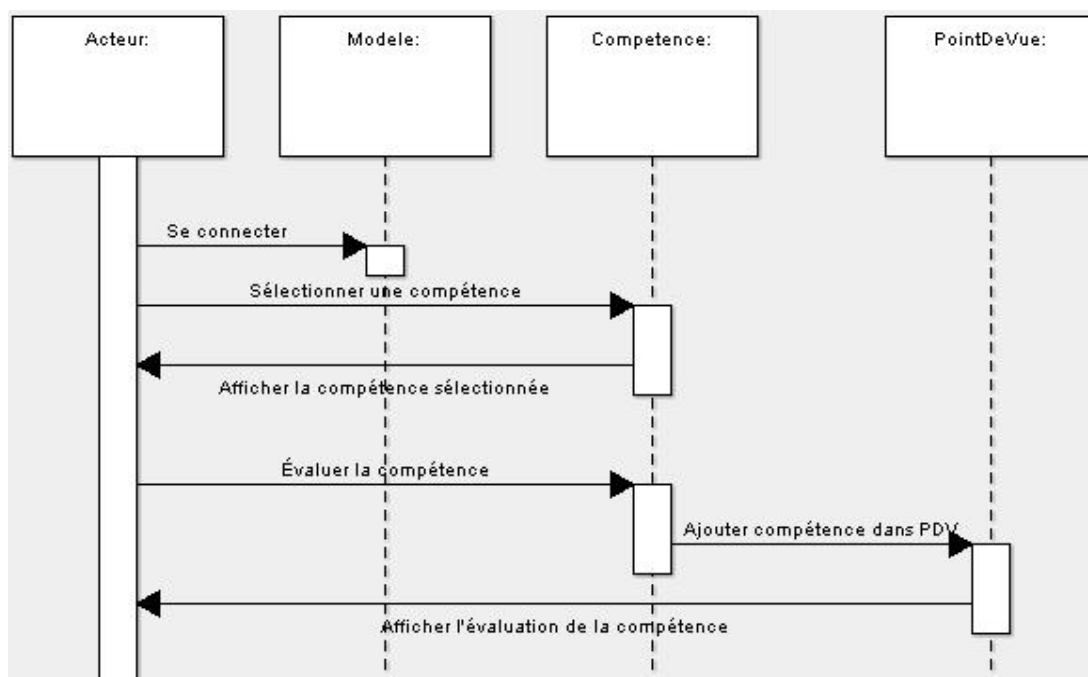


Figure IV-8. Diagramme de séquence Ajouter/Évaluer une compétence

Après s'être connecté au modèle dont il veut évaluer une compétence, l'acteur sélectionne la compétence en question. Celle-ci est affichée à l'écran et l'acteur peut ainsi saisir la valeur de son évaluation. La compétence est ensuite ajoutée au point de vue de l'acteur et affichée à l'écran avec la valeur de l'évaluation.

Le scénario 3 est intitulé *Vérifier une compétence* (voir la figure IV-9). Il consiste à l'interrogation du modèle pour savoir si telle compétence est présente avec tel niveau (ou un niveau supérieur).

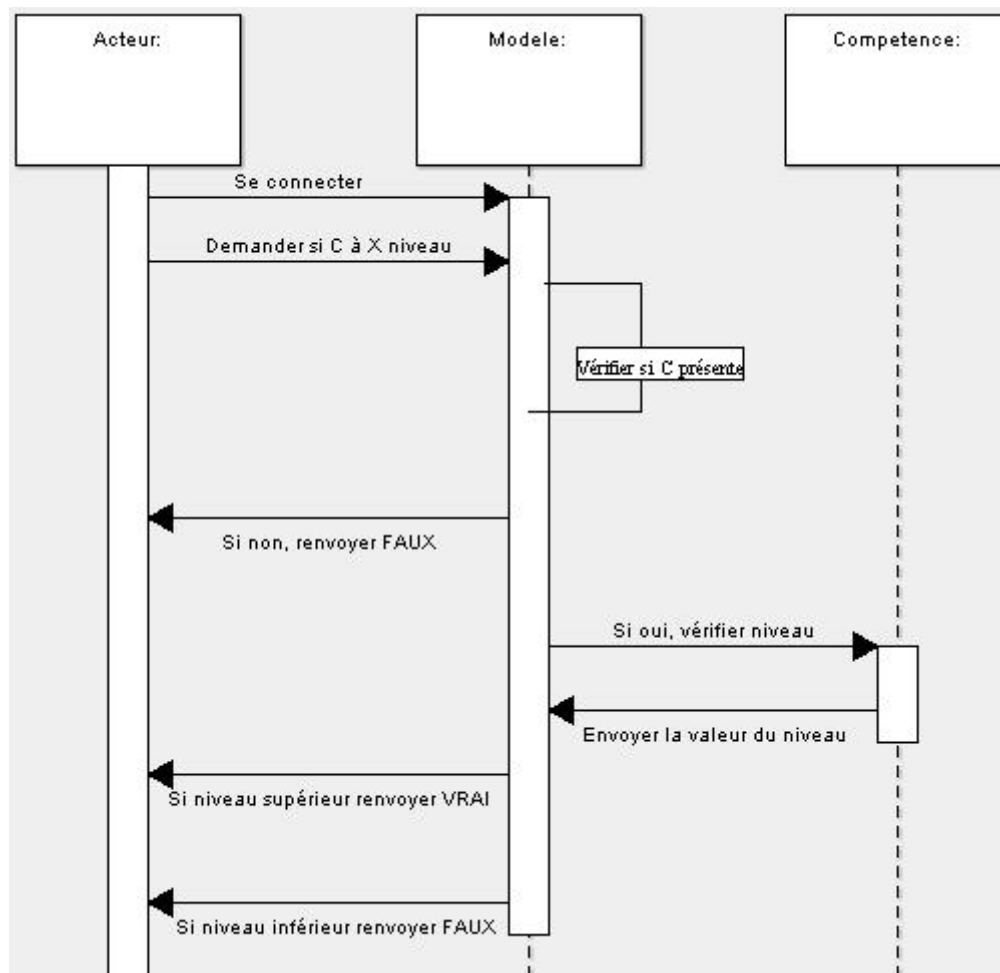


Figure IV-9. Diagramme de séquence Vérifier une compétence

Après s'être connecté, l'acteur formule sa requête à Nosma. Le modèle vérifie si la compétence est présente dans le modèle de l'apprenant principal, si non, il envoie la réponse FAUX à l'acteur. Si la compétence est bien présente dans le modèle, on vérifie son niveau. Si le niveau est égal ou supérieur à celui de la requête, la réponse est VRAI, si le niveau est inférieur, la réponse est FAUX.

Le scénario 4, *Illustrer une compétence par une production*, est représenté par le diagramme de séquence de la figure IV-10. Ceci correspond à ajouter un lien entre une compétence et une production.

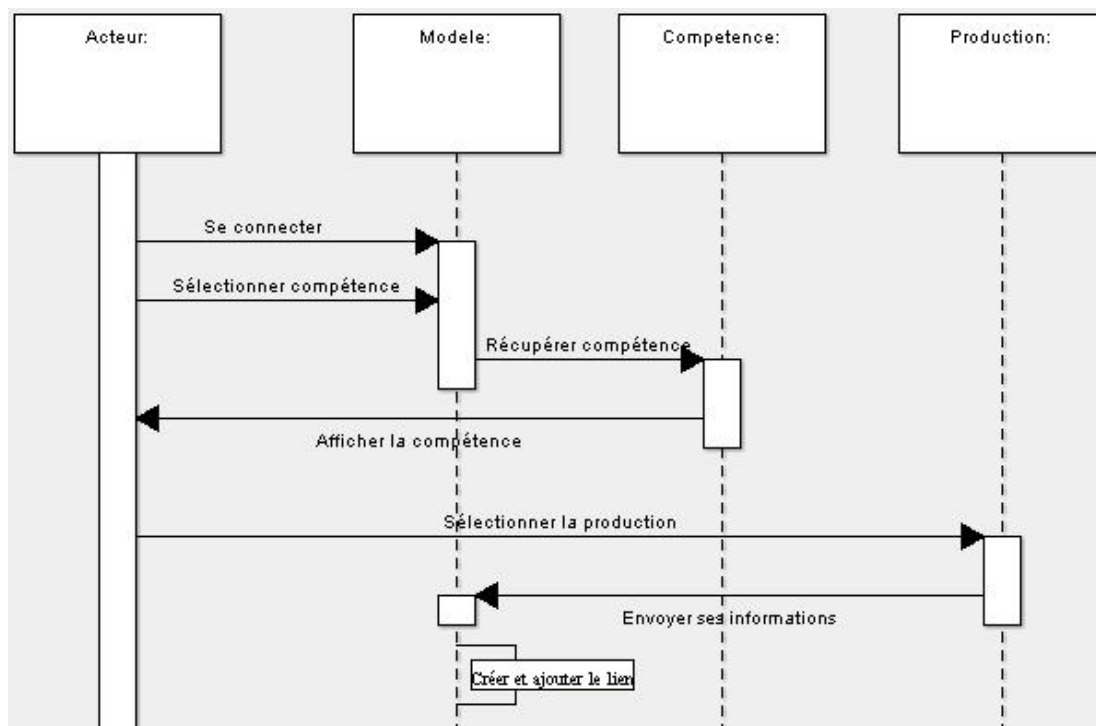


Figure IV-10. Diagramme de séquence Illustrer une compétence par une production

Après s'être connecté au modèle, l'acteur qui illustre la compétence sélectionne celle-ci dans le modèle. Il sélectionne ensuite la production qui illustrera la compétence et le système crée et ajoute le lien dans le modèle de l'apprenant principal ou point de vue consensus (ainsi que l'auteur du lien et le contexte dans lequel il a été ajouté).

Le scénario 5 *Accréditer/tamponner une production* (voir la figure IV-11). Comme expliqué au chapitre III, une production est accréditée lorsqu'elle est de bonne qualité, par exemple, un professeur peut choisir d'accréditer les trois meilleurs projets de la session.

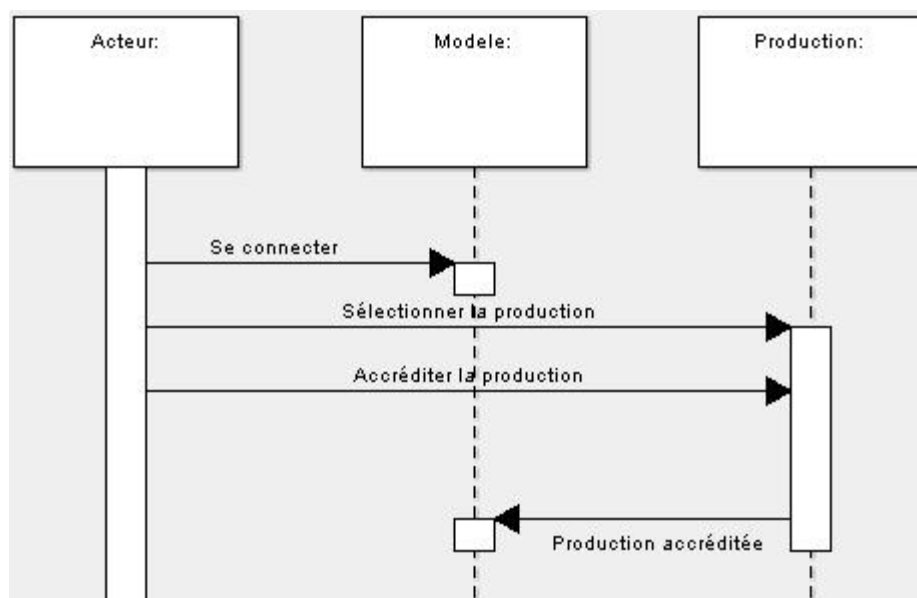


Figure IV-11. Diagramme de séquence Accréditer une production

Après s'être connecté au modèle, l'acteur sélectionne la production concernée et l'accrédite. Seuls les acteurs ayant des droits d'accès au modèle peuvent effectuer ce scénario (selon la spécification de l'élément *droit des acteurs* du contrat associé au modèle contexte temporaire). La production est maintenant accréditée dans le modèle de l'apprenant. Le scénario *Certifier un lien* est similaire à ce scénario.

4.3. Implémentation de Nosma

Dans cette section, nous commençons par présenter les outils que nous avons utilisés pour implémenter Nosma. Principalement, nous présentons Castor, un outil qui nous a permis de transformer des fichiers XML en objets Java et inversement. Nous verrons aussi pourquoi cette transformation nous est utile. Nous présentons aussi Subversion, le système de gestion de version que nous avons utilisé pour l'implémentation de Nosma. Ensuite nous présentons l'implémentation de Nosma, des schémas de son fonctionnement ainsi que ses fonctionnalités. Finalement, nous exposons comment nous avons branché notre système de modélisation de l'apprenant à un système d'apprentissage en ligne existant.

4.3.1. Outils utilisés

4.3.1.1. Objets Java et XML avec Castor

Le cœur de notre système est développé en Java, nous avons ainsi travaillé dans un environnement JRE avec IntelliJ IDEA (c'est un environnement de développement intégré, centré code et axé sur la productivité du développeur). Comme nous l'avons précisé dans le chapitre II, nous avons aussi utilisé l'outil Subversion pour gérer les versions de nos modèles d'apprenant.

Le cœur du système en Java interagit avec les modèles de l'apprenant qui sont stockés en XML afin de favoriser leur interopérabilité. Ainsi, ils peuvent plus facilement être utilisés par d'autres systèmes. En effet, nous souhaitons que notre système puisse être branché à différents systèmes d'apprentissage en ligne. Si nos modèles d'apprenant sont en XML, il sera plus facile de les brancher à ces différents systèmes d'apprentissage et d'échanger des données avec ceux-ci. Dans ces conditions, nous avons cherché un outil nous permettant de « traduire » nos objets Java en XML. Nous avons découvert Castor², un *framework* sous licence libre de *data binding* (lien de données) pour Java. Castor permet de « faire des liens », de passer des objets Java à des documents XML et à des bases de données relationnelles. Nous nous intéressons plus particulièrement à la possibilité de lier des objets Java et des documents XML. Il existe d'autres façons de gérer du XML en Java, les deux principales API sont DOM (*Document Object Model*) et SAX (*Simple Api for XML*). Mais contrairement à ces deux API qui doivent gérer la structure des documents XML, Castor permet de gérer les données contenues dans un fichier XML selon la façon dont elles sont définies dans des classes Java (et inversement). Autrement dit, Castor permet de transformer des données contenues dans un modèle d'objets Java en un document XML. Dans notre cas, un modèle d'objets Java existe déjà (c'est le modèle conceptuel du modèle de l'apprenant), Castor pourra alors être utilisé pour

² <http://www.castor.org/index.html>

faire le pont entre ce modèle d'objets et un document XML. Lorsqu'un modèle de l'apprenant conservé sous forme de XML sera utilisé par notre système, il sera d'abord transformé en objets Java. Ainsi, il sera possible d'effectuer des opérations Java sur les données puis, une fois les opérations nécessaires effectuées, on pourra retransformer les objets Java en fichier XML pour stocker le modèle de l'apprenant.

Pour cela, il est nécessaire d'utiliser un fichier de *mapping* qui permet à Castor de savoir comment faire la transformation, c'est-à-dire un fichier contenant la traduction du document XML vers les objets Java. Il existe des fonctions par défaut que Castor peut utiliser, mais lorsqu'un modèle d'objets Java existe, il est plus bénéfique de créer notre propre fichier de *mapping*. Cela permet de dissocier les changements apportés à la structure du modèle d'objet indépendamment des fichiers XML utilisés. Plus précisément, ce fichier de *mapping* est un fichier XML adoptant le point de vue du modèle d'objet Java et précisant comment se modèle se traduit en XML. Ce fichier doit être conçu de façon à ce que Castor puisse aussi faire le raisonnement inverse (passer du XML aux objets Java) sans ambiguïté. Le fichier de *mapping* que nous avons créé se trouve à l'appendice G page 214.

Concrètement, une classe Java devient un élément de XML. Lorsque Castor transforme un objet Java, la première étape va être de trouver à quel élément correspond la classe Java de l'objet. S'il n'y en a pas, Castor crée par défaut un élément ayant le même nom que la classe Java. Ensuite, Castor utilise les informations contenues dans le fichier de *mapping* pour déterminer comment un attribut de l'objet Java doit être traduit en XML. Un attribut d'objet devient en XML une et une seule de ces possibilités : un attribut, un élément, du contenu textuel ou rien du tout (c'est-à-dire qu'il n'est pas pris en compte, qu'il n'est pas traduit). Cette étape est récursive, c'est-à-dire que si Castor trouve un attribut qui est une classe Java spécifiée ailleurs dans le fichier de *mapping*, il traduira l'objet selon les spécifications de sa classe d'appartenance. Il existe aussi des règles par défaut pour la traduction des attributs d'objet : les attributs de types primitifs deviennent des attributs XML et tous les autres objets deviennent des éléments XML.

Inversement, lorsque Castor utilise le fichier de *mapping* pour traduire un document XML en objets Java, il commence par chercher à quel objet Java correspond un élément XML, s'il ne trouve pas l'information, il crée une classe Java du même nom que l'élément XML. Pour les types primitifs, Castor cherche d'abord un attribut puis un élément XML. Sinon, Castor cherche d'abord un élément puis un attribut.

L'avantage de travailler avec Castor plutôt qu'avec DOM ou SAX est qu'il y a très peu d'apprentissage à faire. Le *data-binding* XML de Castor est très simple à utiliser tout en étant un outil puissant permettant de travailler avec des objets Java et des documents XML, c'est pourquoi nous l'avons utilisé dans le cadre de cette recherche.

4.3.1.2. Subversion (SVN)

Subversion (SVN) est un logiciel de gestion de versions sous licence libre, conçu pour améliorer CVS, le logiciel de gestion de versions le plus reconnu et utilisé. Tout comme CVS, SVN fonctionne avec un serveur centralisé accessible par différents protocoles réseaux (HTTP/HTTPS, SSH...). Les concepteurs de SVN l'ont construit dans l'idée d'améliorer et de simplifier certaines tâches fastidieuses et peu intuitives avec CVS (Collins-Sussman, Fitzpatrick et Pilato, 2002). Ces auteurs considèrent que le modèle de CVS est le bon (entre autre, système centralisé) mais que son opérationnalisation peut être améliorée. Au contraire de LibreSource, Subversion n'est pas une plateforme complète, c'est un logiciel de gestion de versions. Si on choisit d'utiliser cet outil, nous devons le « lier » à notre modèle de l'apprenant pour que les deux fonctionnent ensemble. Nous allons présenter brièvement le fonctionnement de SVN. Pour cela, nous partirons de son architecture (voir figure IV-12).

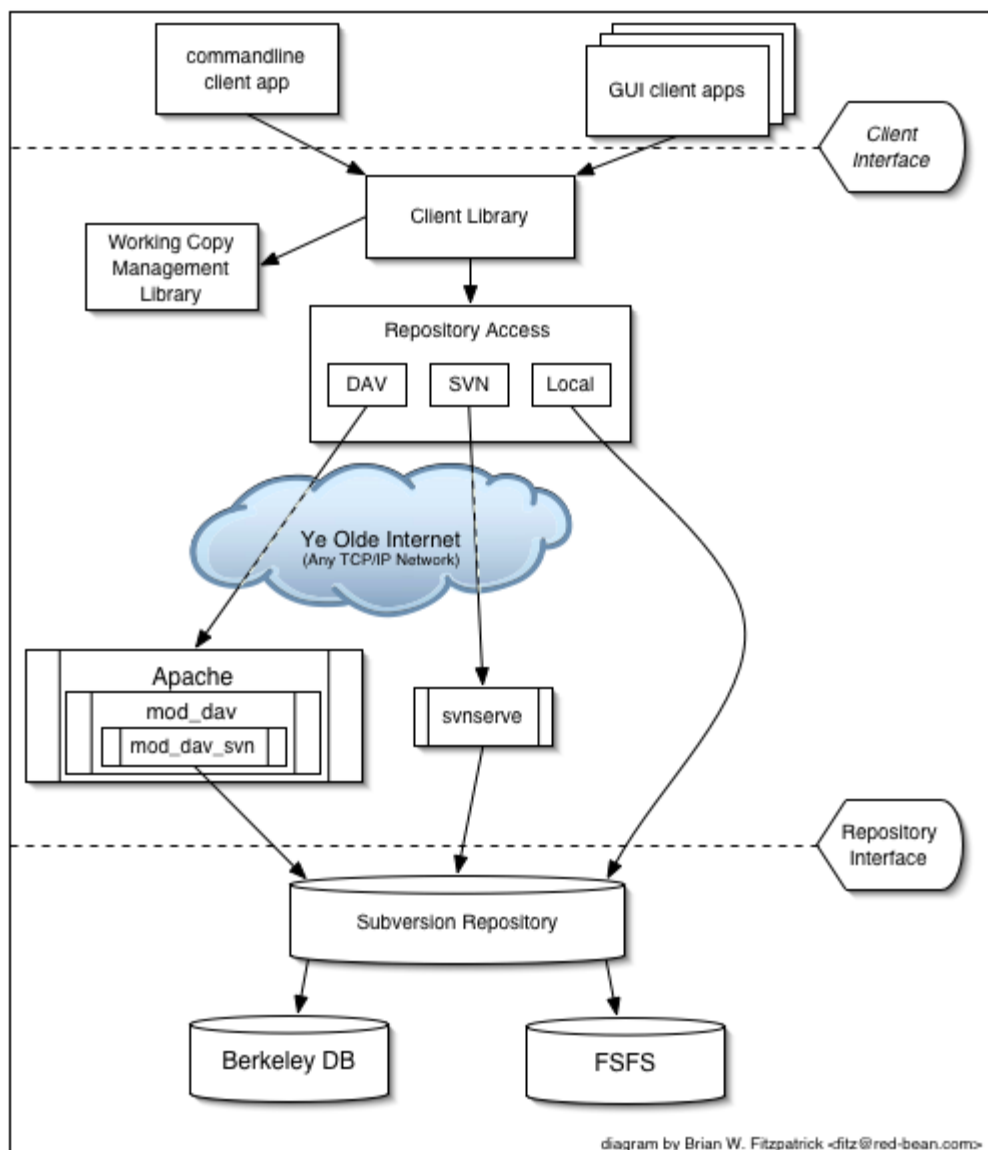


Figure IV-12. Architecture de Subversion (d'après Collins-Sussman *et al.* 2002)

D'un côté, nous trouvons un répertoire SVN qui contient toutes les données qui doivent être versionnées. De l'autre, nous trouvons un programme client qui permet de travailler sur des copies des données (nommées copie de travail). Entre les deux, il existe différents chemins comme nous le voyons sur la figure. L'utilisateur peut interagir avec le serveur par une interface locale qui peut être un logiciel autonome ou l'extension d'un logiciel (comme un logiciel de développement, l'explorateur de

fichier...), en lignes de commande ou de façon graphique. L'utilisateur se connecte au serveur et récupère localement une copie du fichier sur lequel il veut travailler, le modifie puis soumet la nouvelle version au serveur. Si le fichier a été modifié par un autre utilisateur en même temps, un avertissement est produit et SVN crée un fichier tenant compte de toutes les modifications. S'il y a conflit, l'utilisateur doit le résoudre.

Voyons maintenant quels sont les principaux apports de SVN afin de combler certaines faiblesses de CVS. Tout d'abord, les soumissions de modifications sont atomiques, c'est-à-dire qu'elles sont soit refusées soit acceptées dans leur globalité. Subversion permet de renommer et de déplacer des fichiers et des répertoires sans en perdre l'historique. Il est aussi possible de définir des permissions sur un fichier ou un répertoire. De plus les métadonnées (droit d'accès, commentaire...) et les répertoires sont versionnés. Pour finir, les numéros de révision sont globaux, c'est-à-dire que l'ensemble d'une même révision aura le même numéro (alors que dans CVS, chaque fichier obtenait un numéro différent).

Comme nous l'avons dit dans le chapitre II, nous avons utilisé Subversion pour les raisons suivantes :

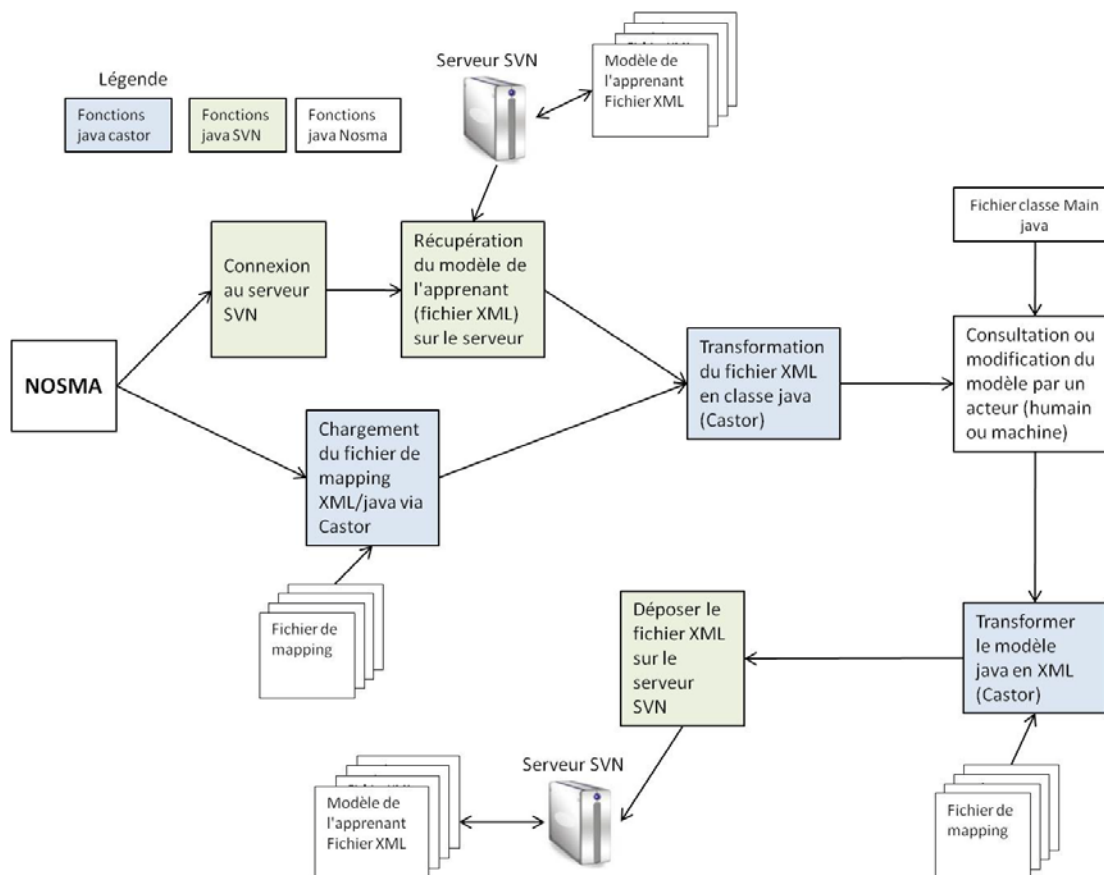
- Subversion est multiplateformes et sous licence libre, et nous pouvons ainsi aisément l'utiliser;
- Subversion fonctionne sur le mode client/serveur et nous pensons qu'il est préférable dans notre recherche d'avoir un dépôt centralisé;
- Subversion semble être le nouveau CVS et en voie de devenir un des outils de gestion de versions les plus utilisés;
- Subversion dispose d'une API Java qui nous a permis facilement de l'utiliser avec notre modèle de l'apprenant.

Cette API Java, SVNKit est sous licence libre et implémente toutes les fonctionnalités de Subversion. SVNKit permet de travailler sur les copies de travail de Subversion, d'accéder et de manipuler le répertoire Subversion depuis une application Java. En utilisant cette API, nous pourrions facilement manipuler et versionner les modèles de l'apprenant puisque notre système est développé en Java.

4.3.2. Implémentation et fonctionnalités

4.3.2.1. Schéma de fonctionnement

Dans cette section, nous allons expliquer le fonctionnement de Nosma et l'interaction entre les différents éléments du système. Nosma a été développé en Java et utilise Castor pour transformer des fichiers XML en objets Java. En effet, les modèles d'apprenant sont stockés sous forme de fichiers XML, pour pouvoir les manipuler, il faut donc les transformer en Java pour que Nosma puisse les traiter. De plus, les fichiers XML sont stockés sur un serveur SVN qui permet de gérer les différentes versions des fichiers (comme expliqué dans le chapitre II). La figure IV-13 représente le modèle procédural du fonctionnement de Nosma.



Comme l'indique la légende, les éléments blancs représentent notre système Nosma, les éléments bleus Castor et les éléments verts SVN. Notre système Nosma se connecte au serveur SVN pour récupérer le modèle de l'apprenant (fichier XML), en parallèle, Castor est utilisé pour s'initialiser à partir du fichier de *mapping*. Une fois ces deux interactions effectuées, le modèle de l'apprenant stocké sous forme de fichier XML est transformé en objets Java et peut ainsi être manipulé par Nosma, par l'intermédiaire d'un acteur (humain ou machine) qui demandera à consulter ou à modifier le modèle. Une fois que l'acteur a terminé d'interagir avec Nosma, le modèle de l'apprenant est retransformé en fichier XML par l'entremise de Castor puis le fichier est déposé sur le serveur SVN (qui gère en même temps le versionnage du fichier). Plus précisément, lorsque le fichier XML est déposé sur le serveur SVN, une nouvelle version du fichier est créée et l'ancienne version est archivée.

Nous présentons maintenant le modèle structurel (voir la figure IV-14), sur lequel nous pouvons voir les interactions entre les éléments de Nosma et avec les utilisateurs.

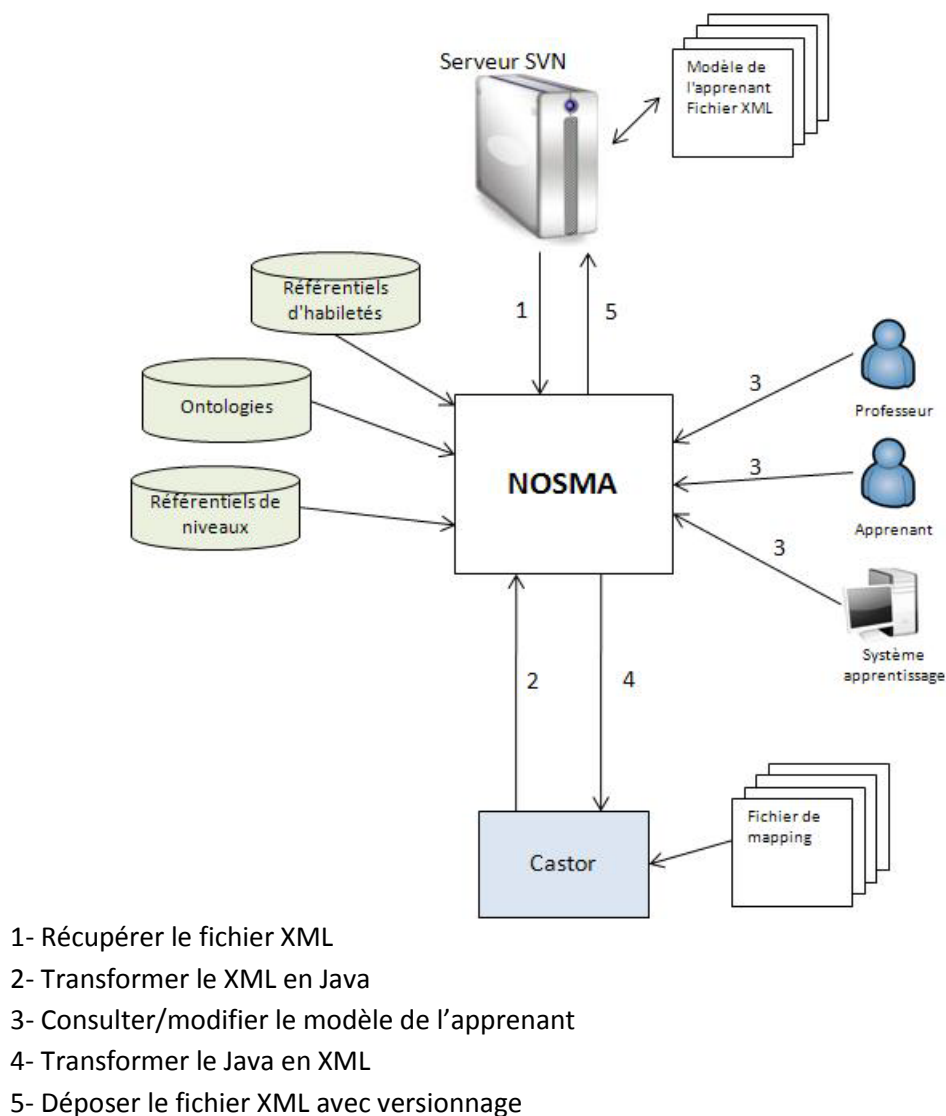


Figure IV-14. Modèle structurel

Comme précisé précédemment, la première étape consiste à la récupération par Nosma du fichier XML sur le serveur SVN. Ensuite en interagissant avec Castor, le fichier est transformé en objets Java. Les utilisateurs peuvent ensuite interagir avec Nosma pour effectuer des opérations sur le modèle de l'apprenant, opérations de consultation ou de modification. Une fois que les utilisateurs ont terminé de manipuler le modèle de l'apprenant, celui-ci est retransformé en fichier XML par

l'entremise de Castor. Finalement, le fichier XML est question est déposé sur le serveur SVN à des fins de stockage et de versionnage (gestion de versions).

Le serveur SVN utilisé est celui du centre de recherche LICEF déjà en place à la création de Nosma. Un répertoire spécifique a été créé pour notre projet de recherche. Aussi, nous avons développé une feuille XSL qui permet d'améliorer l'affichage des fichiers XML (voir appendice H page 221).

4.3.2.2. Fonctionnalités

Nous présentons dans cette section les principales fonctionnalités de Nosma. D'abord, nous parlerons de fonctionnalités systèmes et ensuite de fonctionnalités orientées utilisateur.

Pour que Nosma fonctionne tel que présenté dans les schémas précédents, nous avons dû implémenter des fonctions permettant la communication avec Castor d'une part et avec le serveur SVN d'autre part. Plus précisément, nous avons implémenté une fonction *recupérerModèle* qui permet, comme son nom l'indique, de récupérer un modèle d'apprenant sur le serveur SVN. Cette fonction utilise des fonctions issues de la librairie de Castor pour transformer le modèle récupéré en Java afin que Nosma puisse effectuer des opérations sur le modèle. Elle utilise aussi bien évidemment des fonctions de l'API Java de SVN qui permettent de communiquer avec le serveur à partir du code Java de Nosma. Nous avons aussi implémenté une fonction *déposerModèle* qui permet quant à elle de déposer le modèle sur le serveur une fois que nous ne travaillons plus sur celui-ci. Elle fait, elle aussi, appel à des fonctions des librairies Castor et SVN.

Concernant les fonctionnalités orientées utilisateur, elles se regroupent dans quatre grandes catégories :

- Ajouter dans le modèle
- Supprimer du modèle
- Négocier une compétence
- Interroger le modèle

Les fonctionnalités de la première catégorie sont : ajouter un point de vue, une compétence (dans un point de vue ou dans le modèle principal), une production, une expérience professionnelle, un lien compétence-production, un lien compétence-expérience professionnelle, une certification sur une production ou sur un lien. Concernant la seconde catégorie, il est possible de supprimer des éléments du modèle : supprimer une compétence de tous les points de vue dans lesquels elle est présente, supprimer une compétence du modèle principal, supprimer un point de vue, une production... En ce qui concerne la négociation d'une compétence, nous avons implémenté les trois stratégies présentées précédemment. Finalement, nous avons implémenté deux fonctions d'interrogation du modèle. La première fonction vérifie si une compétence est présente ou non dans le modèle à un certain niveau (ou à un niveau supérieur). La seconde renvoie le niveau associé à une compétence particulière.

Comme mentionné au cours de la présentation de notre modèle conceptuel et de Nosma, certains éléments n'ont pas été implémentés, comme les droits des acteurs, les contrats d'interaction, le métamodèle et les modèles contextes. Ce sont des éléments qu'il serait intéressant d'implémenter dans une future version de Nosma.

Comme nous l'avons précisé, les modèles d'apprenant sont stockés sur un serveur SVN, donc à l'extérieur de notre système Nosma. Ainsi, ils pourraient être utilisés par d'autres systèmes, comme des systèmes d'apprentissage en ligne par exemple. D'autres systèmes pourraient se brancher au serveur SVN pour interagir avec un modèle (pour y ajouter des informations ou pour le consulter dans le but par exemple d'adapter un scénario pédagogique). C'est ce que nous explorons dans le paragraphe suivant.

4.3.3. Implémentation d'un lien avec un SAL : TELOS

4.3.3.1. Le connecteur de TELOS

Nous avons étudié la possibilité de brancher Nosma à un système d'apprentissage en ligne. Plus précisément, nous avons étudié la possibilité de brancher Nosma à TELOS (*Tele-learning operating System*), qui est développé au centre de recherche LICEF. TELOS vise à fournir un système d'opération ouvert (*open operating system*) dans lequel les utilisateurs peuvent développer et utiliser des environnements d'apprentissage en ligne et de gestion des connaissances dans un *framework* orienté services général et intégré. TELOS est principalement fondé sur un ensemble de fonctionnalités de coordination et de synchronisation qui supportent les interactions entre personnes et ressources informatisées composant un système d'apprentissage ou de gestion des connaissances (Magnan et Paquette, 2007). TELOS comporte la possibilité d'ajouter un composant externe afin de permettre aux deux systèmes d'interagir. Les composants externes peuvent être vus comme des bibliothèques externes qui offrent des services indépendants mais ils peuvent aussi se comporter comme des modules internes de TELOS. Pour permettre cette interaction, TELOS dispose d'un connecteur qui permet de faire le lien entre TELOS et le composant externe. Plus précisément, le connecteur permet de lancer un composant externe, d'appeler les fonctions publiques de ce composant et d'offrir au composant les services publics de TELOS. Actuellement, le connecteur permet des composants externes développés en Java. Comme Nosma est développé en Java, ce connecteur pour composants externes semblent tout à fait convenir à notre projet. La figure IV-15 montre comment TELOS et les composants externes communiquent.

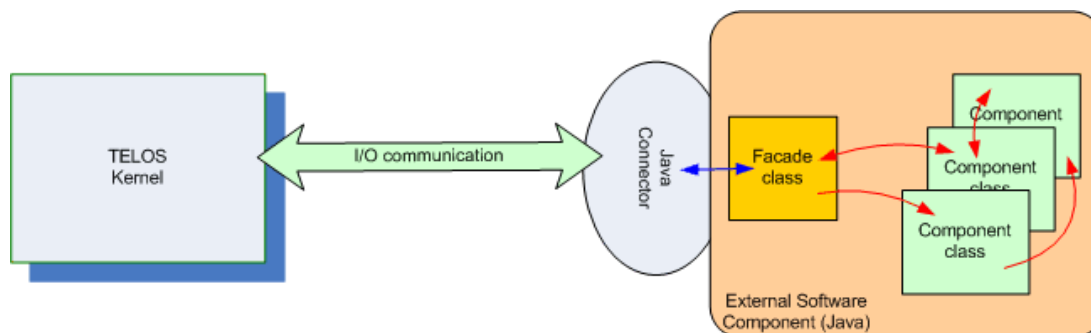


Figure IV-15. Interaction entre TELOS et les composants externes (tiré de Miara, 2010)

Comme nous le voyons sur la figure ci-dessus, le connecteur interagit avec une classe façade du composant externe, qui elle-même communique avec les autres classes du composant. Pour que l'interaction fonctionne, le composant externe doit être « compatible » avec TELOS, par exemple, il doit comporter une classe façade. Plus précisément, TELOS impose quatre conditions pour qu'un composant externe puisse interagir avec un connecteur Java (en plus d'être développé en langage de programmation Java) :

- (1) le composant externe doit posséder une classe façade;
- (2) les méthodes de la classe façade utilisables par TELOS doivent être publiques;
- (3) les méthodes ne peuvent pas comporter plus de cinq arguments;
- (4) les arguments doivent être d'un des types acceptés par TELOS (soit String, int, double, long, boolean et vector).

Le connecteur offre aussi la possibilité au composant externe d'appeler l'API de TELOS. Ainsi, il est possible d'appeler des fonctions de TELOS dans les fonctions du composant externe. Nous n'avons pas eu besoin d'exploiter cette possibilité dans le cadre de notre recherche. Pour que ces liens et interactions soient possibles, il faut télécharger le composant externe dans TELOS et créer un manifeste qui précise, entre autre, le nom de la classe façade. Une fois le composant externe importé dans TELOS avec son manifeste, il est possible d'utiliser les fonctions du composant dans TELOS. Par exemple, dans notre cas, au cours d'un scénario pédagogique, il serait possible d'interroger le modèle de l'apprenant pour savoir si

telle compétence y est présente. Nous allons préciser plus en détail les fonctions que nous avons développées dans TELOS après avoir présenté les modifications que nous avons dû apporter à Nosma pour qu'il soit compatible avec TELOS et qu'il puisse ainsi être importé en tant que composant externe.

4.3.3.2. **Rendre Nosma compatible avec TELOS**

Pour que Nosma soit importable en tant que composant externe dans TELOS, il doit répondre aux conditions de TELOS, soit comporter une classe façade avec des fonctions publiques qui n'ont pas plus de cinq arguments qui sont d'un des types acceptés par TELOS. Nous avons donc créé une classe *Manager* qui sert de classe façade pour l'interaction avec le connecteur Java de TELOS. Dans cette classe, nous avons implémenté des fonctions qui respectent les conditions de TELOS et qui font appel aux fonctions déjà existantes de Nosma (présentées précédemment au paragraphe 4.3.2.2. Fonctionnalités). Plus précisément, nous avons implémenté cinq fonctions pour illustrer la faisabilité de l'interaction entre Nosma et TELOS (qui sera démontrée dans le chapitre V avec le scénario 3). Tout d'abord, nous avons implémenté les fonctions d'accès au répertoire SVN soit *recupererModele* et *deposerModele*. Ces deux fonctions contiennent en paramètre l'acteur dont on cherche le modèle et l'acteur qui cherche le modèle. Pour ne pas accéder au répertoire SVN plusieurs fois au cours d'une même session, nous avons ajouté à la classe façade un tableau contenant tous les modèles d'apprenant déjà consultés au cours de la session. À chaque fois que le composant externe est relancé (à partir de TELOS), ce tableau est à nouveau vide et se remplit au fur et à mesure de l'utilisation de Nosma depuis TELOS. Ainsi, avant de se connecter au serveur SVN pour y récupérer un modèle d'apprenant spécifique, la fonction *recupererModele* vérifie si le modèle n'est pas déjà dans le tableau. S'il ne l'est pas, la connexion au serveur SVN a lieu et le modèle est ajouté dans le tableau pour utilisation future (au besoin). La connexion au serveur SVN est réalisée par l'entremise de la fonction déjà développée dans Nosma

(présentée au paragraphe 4.3.2.2.). De même, *deposerModele* fait appel à la fonction déjà existante de Nosma. Nous avons aussi implémenté deux fonctions de modification du modèle et une fonction d'interrogation du modèle, soit *ajouterCompetence*, *ajouterProduction* et *hasCompetence* qui permettent d'ajouter une compétence et une production et de savoir si la compétence X est dans le modèle au niveau minimum Y. Ces fonctions font appel aux fonctions existantes de Nosma et ne comportent que cinq paramètres de types acceptés par TELOS.

4.3.3.3. Utilisation des fonctions Nosma dans TELOS

TELOS permet de créer des scénarios pédagogiques graphiquement. Ces scénarios pédagogiques peuvent comporter des fonctions et des activités. Nous avons créé des fonctions dans TELOS utilisant les fonctions de notre classe façade. Ainsi, Nosma peut être utilisé dans les scénarios pédagogiques créés dans TELOS. Nous verrons dans le chapitre V un exemple d'un tel scénario.

Nous avons créé quatre fonctions dans TELOS : *déposerModèle*, *ajouterCompétence*, *ajouterProduction* et *hasCompétence*. La fonction *recupererModele* n'est pas développée parce qu'elle est incluse dans les fonctions de modification et d'interrogation du modèle de l'apprenant. Présentons plus en détail la fonction *ajouterProduction* de TELOS (voir la figure IV-16).

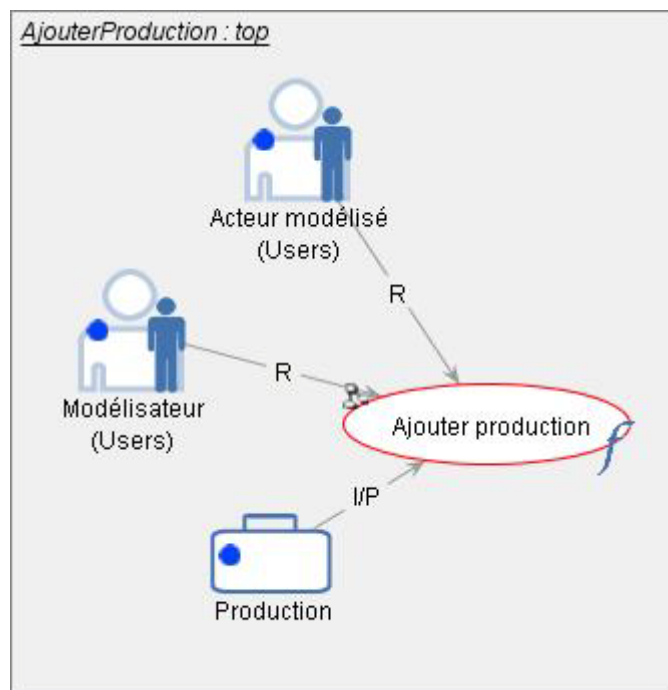


Figure IV-16. Fonction Ajouter production de TELOS

Cette figure montre que cette fonction a trois arguments, deux acteurs (l'acteur dont c'est le modèle, soit l'acteur modélisé et l'acteur interagissant avec le modèle, soit le modélisateur) ainsi qu'un document qui est la production à ajouter.

Le signe en haut à gauche de l'ovale *ajouterProduction* signifie que la fonction comporte un sous-niveau. Il est présenté à la figure IV-17. Dans cette fonction, nous faisons appel à des fonctionnalités de TELOS pour récupérer des informations sur les éléments de TELOS, comme par exemple le titre du document que nous voulons ajouter au modèle de l'apprenant. Nous avons besoin de ce titre pour créer la fiche de métadonnées de la production qui sera ajoutée au modèle de l'apprenant.

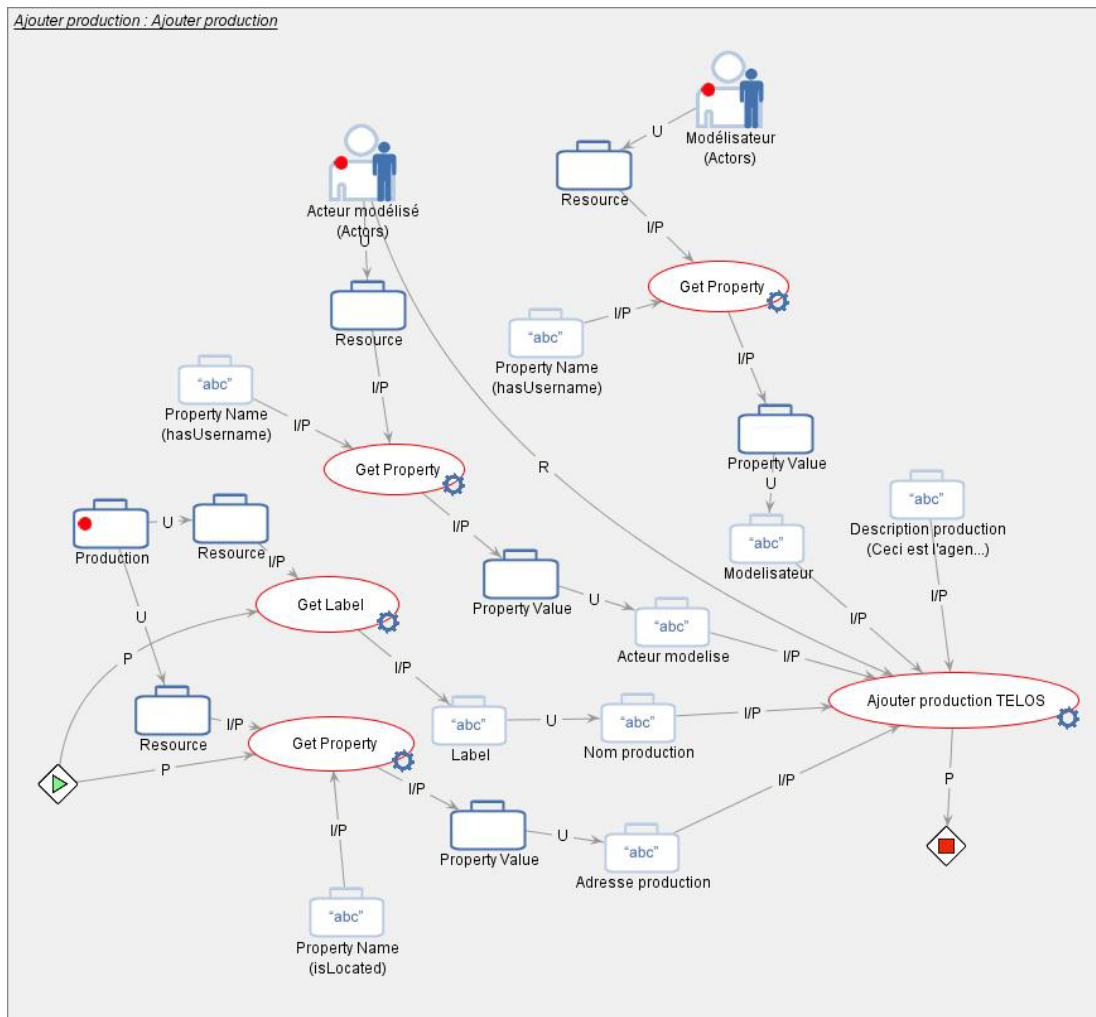


Figure IV-17. Sous-niveau de la fonction Ajouter production de TELOS

Expliquons plus en détail la figure. La flèche verte, à gauche, signifie le début et le carré rouge la fin. La fonction débute par les fonctions *getLabel* et *getProperty*, ce sont des fonctions de TELOS qui permettent d'obtenir des informations sur les éléments de TELOS. Ces fonctions nous permettent d'interroger TELOS pour connaître l'étiquette du document (soit le nom du fichier) entré en paramètre de la fonction *ajouterProduction* et l'adresse du document. Nous avons besoin de ces informations pour créer la fiche de métadonnées de la production qui sera ajoutée au modèle de l'apprenant. Les autres fonctions *getProperty* permettent de récupérer les noms des acteurs de TELOS. Toutes ces informations récupérées sur les éléments de

TELOS sont ensuite données en argument de la fonction *ajouterProduction TELOS* qui est une fonction de Nosma et qui fait appel à la fonction *ajouterProduction* de la classe façade *Manager* que nous avons implémentée spécialement pour communiquer avec TELOS.

Des traitements similaires sont effectués dans les autres fonctions combinant les fonctionnalités de TELOS et de Nosma. Le dernier scénario de simulation présenté dans le chapitre V montre une utilisation des fonctions précédentes dans un scénario d'apprentissage.

4.4. Conclusion

Nous avons débuté ce chapitre en présentant la méthode appliquée pour le développement de notre prototype. Pour résumer, le prototype a été développé en quatre itérations (itérations correspondant à la phase 2, soit la phase d'élaboration et de construction). Ces itérations sont :

1. Développement de la structure du modèle de l'apprenant
2. Ajout de Castor pour passer des objets Java au XML
3. Ajout de Subversion pour la gestion de versions
 - a. Utilisation de l'API subversion pour versionner des fichiers
 - b. Interaction entre Castor et SVN
4. Développement/affinement du prototype selon les 2 scénarios conçus

Les sections suivantes ont consisté à présenter une partie de l'étape d'analyse. Tout d'abord, nous avons exposé les principaux cas d'utilisation de notre système, le diagramme des cas d'utilisation et des diagrammes d'interaction. Nous avons aussi présenté trois stratégies de négociation que nous avons ensuite implémentées. Ensuite, nous avons présenté plus en détail l'implémentation de Nosma, en commençant par la présentation de deux outils que nous avons utilisés pour cette implémentation. D'abord nous avons présenté Castor, outil que nous avons utilisé

pour passer des objets Java aux fichiers XML et inversement. Puis nous avons présenté Subversion, outil de gestion de versions que nous avons branché à notre implémentation de Nosma. Ensuite, nous avons exposé le fonctionnement architectural de Nosma, soit les interactions entre les différents éléments intervenant dans le système. Nous avons poursuivi avec les fonctionnalités développées dans le cadre de notre recherche. Finalement, nous avons présenté comment nous avons procédé pour brancher Nosma à un système d'apprentissage en ligne, TELOS.

Chapitre V - ÉVALUATION ET VALIDATION

Ce chapitre correspond à la phase 3 intitulée *Transition* de la démarche de recherche présentée en introduction de ce document. Cette phase consiste en tests et correspond à l'évaluation et à la validation de notre travail. Dans un travail de recherche, on peut considérer une évaluation comme un processus systématique de collecte de données appropriées et d'analyse de ces informations afin d'aider à la prise de décision (Cooley et Lohnes, 1976). Dans ce chapitre, nous présentons les résultats de l'évaluation des produits de notre thèse à savoir le modèle conceptuel de modèle de l'apprenant et son opérationnalisation par le prototype Nosma. Nous commençons par expliquer notre choix quant à la méthode d'évaluation appliquée en exposant différentes méthodes et types d'évaluation existants. Ensuite, nous exposons la démarche que nous avons suivie pour effectuer cette évaluation sous forme de trois simulations (ou scénarios de test). Et finalement, nous présentons les résultats de ces simulations.

5.1. Choix d'une méthode d'évaluation

Le principal but de ce chapitre est de valider la pertinence d'un modèle cognitif de l'apprenant, qui évolue dans le temps afin d'assister l'apprentissage tout au long de la vie, qui est ouvert aux différents contextes de vie d'un individu et aux différents acteurs prenant part au processus d'apprentissage. Cette validation se fait sur la base de l'opérationnalisation du modèle par Nosma. Ainsi, pour valider notre modèle conceptuel, nous évaluons Nosma. Plus précisément, l'évaluation de Nosma consiste à s'assurer de l'efficacité de ce système à répondre à son cahier des charges, ou à

s'assurer qu'il fasse effectivement et efficacement ce pour quoi il a été conçu. Ce type d'évaluation est appelée *preuve d'exactitude* (Iqbal *et al.*, 1999). Dans les sections qui suivent, nous allons détailler les méthodes d'évaluation à notre disposition et préciser pourquoi nous avons choisi de valider notre travail en évaluant le prototype Nosma.

Dans le domaine des EIAH, il existe peu de consensus sur les moyens à utiliser pour évaluer ceux-ci. Cependant, Mark et Greer (1993), dans leur étude approfondie sur les méthodes d'évaluation des EIAH, ont mis en évidence que des disciplines connexes (développement des systèmes experts, *computer-based instruction*, éducation, informatique, génie et psychologie) ont développé des méthodes d'évaluation qui peuvent être applicables aux EIAH. Ces méthodes recensées et classifiées par Iqbal *et al.* (1999) ont été utilisées par plusieurs chercheurs du domaine des EIAH.

Deux questions se posent au commencement d'un processus d'évaluation : « Comment évaluer ? » et « Quand évaluer ? » (Tricot *et al.*, 2003). Les réponses à ces questions déterminent respectivement : (1) le choix de la méthode et (2) le type d'évaluation. Concernant le choix de la méthode, il s'agit de décider entre une méthode quantitative et une méthode qualitative. Concernant le type d'évaluation, il s'agit de décider entre une évaluation interne ou une évaluation externe. Les sections suivantes traitent de ces deux questions.

5.1.1. Méthode d'évaluation

L'approche quantitative est appropriée lorsque l'on désire évaluer une hypothèse ou une théorie élaborée par l'observation de phénomènes. Elle consiste à colliger des données numériques relatives à un certain nombre de variables et à analyser ces données en vue d'établir une relation causale ou corrélationnelle et de vérifier ainsi des hypothèses. La méthode quantitative, dont l'analyse statistique est l'exemple typique, est une démarche de quantification de données, provenant d'un

grand nombre de participants, dans le but d'obtenir des résultats reproductibles et généralisables et d'exclure les particularités individuelles et subjectives.

L'approche qualitative vise plutôt à étudier en profondeur un objet ou un phénomène, c'est une démarche de recherche de sens (Paillé et Mucchielli, 2003). Elle consiste à collecter des données qualitatives (paroles, comportements, textes...) sur l'objet ou le phénomène et à analyser ces données afin de le caractériser et d'en trouver ainsi son sens, sa pertinence même. La principale différence avec l'approche quantitative se situe au niveau de l'analyse et des conclusions tirées. Dans une approche qualitative, le chercheur observe afin d'inférer et de caractériser ou alors de vérifier un phénomène. Bien que les résultats qualitatifs ne soient pas nécessairement généralisables, ils sont exploratoires et explicatifs et servent souvent de point de départ pour la validation de nouveaux concepts, la validation de système informatique et finalement pour l'évaluation quantitative (Miles et Huberman, 2003).

Si le but d'une évaluation est de prouver l'efficacité d'un système par rapport à ses fonctionnalités et à ses utilisateurs, alors une évaluation quantitative est souhaitable à la fin du développement du système. Toutefois, certaines recherches s'intéressent à la description, la caractérisation et la justification de la nature profonde des artefacts qu'elles produisent (Murray, 1993). Dans ces cas, les méthodes qualitatives doivent être mises à contribution. De plus, en cours de développement, il est plus bénéfique et moins coûteux d'utiliser des méthodes qualitatives. Nous nous situons exactement dans ce cas de figure puisque nous visons la validation de notre modèle conceptuel de modèle de l'apprenant par l'évaluation de son opérationnalisation par le prototype Nosma (nous ne visons pas uniquement l'évaluation du système Nosma).

5.1.2. Type d'évaluation

Les méthodes d'évaluation proposées dans la littérature sont nombreuses et variées : inspection experte, niveau d'accord, *Wizard-of-Oz*, métrique de performance,

évaluation basée sur des critères, certification, preuve d'existence, observation et classification qualitative, analyses de tâches structurées et classification de phénomènes, études comparatives, etc. (Iqbal *et al.*, 1999, Mark et Greer, 1993 et Murray, 1993). Selon la méthode employée, il est possible d'effectuer une évaluation interne (en cours de conception) ou une évaluation externe du système (en fin de conception et *a posteriori*).

Une évaluation externe (dont le but est de tester le système au complet) est souhaitable si le système à évaluer est déjà au stade de *béta-testing* (on parle d'évaluation en fin de conception ou de validation); ou bien si le système est non seulement conçu mais réalisé (on parle d'évaluation *a posteriori*) (Tricot *et al.*, 2003).

Toutefois, si le système à évaluer est aux premiers stades de conception ou au stade de prototype, alors nous avons besoin d'une évaluation interne qui va valider le processus de conception ou qui va tester des composantes spécifiques du système en cours de conception. Une telle évaluation permet de corriger à moindre frais des problèmes qu'il serait très coûteux voire impossible à corriger en fin de conception et contribue, pour l'équipe de conception, au partage de représentations cognitives communes sur l'artéfact en cours d'élaboration (Tricot *et al.*, 2003). Ceci est notre cas puisque nous désirons évaluer Nosma qui est un prototype. De plus, nous avons choisi une démarche itérative et les résultats de l'évaluation présentée dans ce chapitre serviront d'intrant pour la boucle suivante de conception-réalisation de notre modèle de l'apprenant.

5.1.3. Critères d'évaluation

Nous avons parlé de méthode et de type d'évaluation mais certains parlent plutôt de critères d'évaluation (Nogier, 2008, Nielsen, 2003, Tricot *et al.*, 2003, Mayhew, 1999). L'ISO définit l'utilisabilité (dans la norme ISO 9241-11 :1998) comme « le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction,

dans un contexte d'utilisation spécifié ». Trois critères caractérisent ainsi l'utilisabilité : l'efficacité (le système permet à ses utilisateur d'atteindre le résultat prévu, le système fait ce qu'il est censé faire), l'efficience (concerne l'atteinte du résultat avec un effort moindre et dans un temps minimal) et la satisfaction (concerne la satisfaction de l'utilisateur à interagir avec le système).

Nielsen (1993) distingue utilisabilité et utilité. Pour lui, l'utilité concerne la conformité entre les buts visés par les utilisateurs et les résultats obtenus grâce au système. Mesurer l'utilité d'un système revient à en mesurer sa pertinence, c'est-à-dire si ses fonctionnalités permettent d'atteindre les buts visés lors de sa conception, si le système fait ce pour quoi il a été conçu. L'utilisabilité concerne plutôt la qualité de l'interaction avec l'utilisateur, c'est-à-dire la possibilité pour celui-ci d'effectuer ses tâches de façon conviviale et satisfaisante avec le système. Nielsen intègre dans l'utilisabilité cinq critères : l'efficience (atteindre le but fixé avec moindre effort et temps minimal), l'apprenabilité (ou facilité d'apprentissage et d'appropriation), la mémorisation (mémorisation du fonctionnement du système), la fiabilité (prévention des erreurs) et la satisfaction de l'utilisateur. Scapin et Bastien (1997) ne prennent pas en compte l'efficience dans leurs critères ergonomiques qui sont au nombre de huit (guidage, charge de travail, contrôle explicite, adaptabilité, gestion des erreurs, homogénéité/cohérence, signifiante des codes et dénominations et compatibilité). À cette étape de validation de notre travail, nous visons l'évaluation de l'utilité de Nosma et non encore de son utilisabilité.

Pour résumer, nous utilisons une approche d'évaluation qualitative, un type d'évaluation interne (soit en cours de conception et de développement) et nous appliquons la *preuve d'exactitude* (Iqbal *et al.*, 1999) pour évaluer Nosma, ce qui nous permet de valider notre modèle conceptuel de modèle de l'apprenant. Plus précisément, évaluer l'implémentation de Nosma par *preuve d'exactitude* consiste à s'assurer que le système répond bien aux besoins qui ont été définis lors de sa conception. Pour cela, nous avons testé Nosma à l'aide de trois scénarios associés à des contextes d'utilisation précis. Nous avons appelé ces ensembles scénarios-

contextes « simulations » et c'est le terme que nous utiliserons dans la suite de ce chapitre. Simulation dans le sens de simuler la réalité. En effet, les trois contextes que nous utilisons sont tirés de la réalité et nous avons tenté de concevoir des scénarios aussi proches de la réalité que possible.

Dans les sections suivantes, nous présentons d'abord la démarche que nous avons adoptée pour effectuer les trois simulations avec Nosma puis nous présentons les résultats de ces simulations.

5.2. Démarche

Les simulations présentées dans ce chapitre ont pour but d'évaluer le prototype Nosma. Une simulation constitue une concrétisation particulière d'un modèle. Le fait d'attribuer des valeurs aux variables d'un modèle et d'exécuter le programme constitue une simulation (Van der Maren, 2003). C'est ce que nous avons fait et nous présentons dans ce chapitre les résultats de ces simulations.

Rappelons que nous visons à enrichir le champ des EIAH par la création d'un modèle cognitif de l'apprenant, qui évolue dans le temps afin d'assister l'apprentissage tout au long de la vie, qui est ouvert aux différents contextes de vie d'un individu et aux différents acteurs prenant part au processus d'apprentissage. Pour répondre à cet objectif, le modèle de l'apprenant que nous proposons se doit d'intégrer les connaissances, les compétences et les productions de l'apprenant, d'évoluer dans le temps au même titre que l'apprentissage et de prendre en compte différents points de vue (ceux des acteurs participants au processus d'apprentissage).

Par les simulations présentées dans ce chapitre, nous voulons vérifier que nous avons bien atteint nos objectifs de recherche. Par le biais du prototype développé à partir de notre modèle conceptuel, nous allons donc montrer comment notre modèle permet le référencement sémantique des apprenants représentés grâce à l'utilisation du concept de compétence, comment notre modèle permet le suivi de l'évolution dans

le temps des apprentissages en offrant une gestion de l'évolution (dans le temps) des modèles d'apprenant et comment il permet l'engagement et la prise en compte des acteurs participant au processus d'apprentissage en gérant de multiples points de vue. Nous montrerons aussi que Nosma peut être utilisé dans différents contextes.

Plus précisément, nous effectuons trois simulations chacune dans un contexte différent. Chaque scénario et contexte (constituant une simulation) ont été choisis et conçus afin de démontrer des aspects spécifiques de notre proposition. Ainsi, une simulation ne vise pas à faire la preuve de toutes les caractéristiques de notre modèle de l'apprenant. Nous préciserons dans les paragraphes qui suivent les aspects démontrés par chaque simulation. Nous démontrerons à travers ces trois simulations :

- le bon fonctionnement général de Nosma (ajout de compétence, production, utilisation de point de vue, etc.);
- la fermeture du modèle contexte temporaire (soit la négociation de toutes les compétences et la conservation du point de vue de l'apprenant, et ce, pour les différentes stratégies de négociation conçues);
- l'utilisation du modèle avec une autre vision de la compétence ainsi qu'avec d'autres référentiels que ceux de Paquette;
- l'utilisation de Nosma en interaction avec un système d'apprentissage en ligne.

Dans la suite de ce chapitre, nous présentons les trois simulations ainsi que leurs contextes et scénarios d'utilisation et, bien entendu leurs résultats. La description détaillée des simulations (référentiels de compétence, ontologies, déroulement du scénario, etc.) et le détail des résultats étape par étape (impressions écran à chaque étape) se trouvent à l'appendice I page 226.

5.3. Simulation 1 : cours en ligne avec acteurs humains

Dans cette simulation, nous nous situons dans un contexte similaire au contexte d'un cours universitaire, mais pour les besoins de la simulation, nous modifierons le contexte original dont nous nous inspirons (présenté au paragraphe qui suit). Dans ce

contexte, les professeurs et tuteurs sont les évaluateurs des compétences et les décideurs, c'est-à-dire que la stratégie de négociation utilisée se fonde sur leurs évaluations. C'est la première stratégie décrite au chapitre IV pages 109 et 110.

Dans ce scénario, on veut faire la preuve d'une part du bon fonctionnement de notre système et d'autre part de la fonction de négociation 1. Pour la négociation, nous montrerons celle-ci dans tous les cas de figure pouvant se présenter (voir chapitre IV). Concernant notre système, cette simulation montre le bon fonctionnement de :

- l'utilisation de points de vue
- l'ajout d'une compétence, d'une production
- la preuve d'une compétence par une production
- la certification d'un tel lien
- la fermeture du modèle contexte temporaire (soit la négociation de toutes les compétences et la conservation du point de vue de l'apprenant).

5.3.1. Contexte

5.3.1.1. Description du contexte

Nous nous inspirons du cours universitaire INF 4018 intitulé *Projet d'intégration* fondé sur une activité d'autoévaluation des compétences du professeur Richard Hotte à la Télé-université. Ce cours est un cours-projet permettant aux étudiants de faire la synthèse des connaissances et des habiletés développées au cours du programme (ce cours a été développé pour deux programmes : le certificat en informatique appliquée à l'organisation et le certificat en informatique appliquée). Plus précisément, le cours permet aux étudiants de faire le bilan de leurs compétences techniques, professionnelles et personnelles en vue de les utiliser dans la réalisation d'un projet professionnel (Hotte, 2009). Pour effectuer ce bilan, un référentiel de compétences du programme au complet a été développé. C'est pourquoi nous avons

choisi ce cours comme contexte de notre première simulation, avec l'autorisation du professeur Hotte.

5.3.1.2. **Référentiel de compétences**

Le référentiel de compétences du cours du professeur Hotte a été développé avec la définition de compétence de Paquette (2002a), avec son référentiel d'habiletés génériques et avec ses indicateurs de performance (présentés au chapitre I).

Nous n'allons pas utiliser l'ensemble du référentiel de compétences du cours du professeur Hotte car celui-ci est très complet et nous n'avons pas besoin d'autant de compétences pour effectuer notre simulation. Pour notre simulation, nous avons choisi un ensemble de 11 compétences (nommées C1 à C11) organisées en quatre catégories : 1) compétence technique – générale, 2) compétence professionnelle – communiquer, 3) compétence professionnelle – travailler avec les autres et 4) compétence personnelle. Le tableau qui présente ces compétences se trouve à l'appendice I page 229. Dans notre modèle de l'apprenant, une compétence n'est pas liée à une catégorie mais plutôt à une ontologie. Dans le cadre de cette simulation, nous n'utilisons pas cet aspect du modèle (il sera démontré par une autre simulation).

5.3.1.3. **Référentiels d'habiletés et de niveaux**

Les référentiels utilisés dans ce contexte sont ceux de Paquette (2002a). Rappelons ceux-ci brièvement. Le référentiel d'habiletés comporte 10 habiletés génériques ordonnées par ordre ascendant (1. Porter attention, 2. Intégrer, 3. Instancier/préciser, 4. Transposer/traduire, 5. Appliquer, 6. Analyser, 7. Réparer, 8. Synthétiser, 9. Évaluer, 10. Auto-contrôler).

Le référentiel de niveaux utilisé correspond aux indicateurs de performance de Paquette (2002a). Le référentiel de Paquette comporte cinq indicateurs et quatre niveaux (voir tableau I-2 page 34). Ces quatre niveaux sont du plus bas au plus élevé

sensibilisation (niveau 1), familiarisation (niveau 2), maîtrise (niveau 3) et expertise (niveau 4).

5.3.1.4. Contrat

Comme nous l'avons spécifié au chapitre III (page 85), les contrats d'interaction avec notre modèle de l'apprenant comportent quatre points :

1. **droits des acteurs** : les professeurs et tuteurs intervenant dans le cours ont des droits d'accès au modèle. Ils peuvent évaluer les compétences dont ils sont responsables et consulter l'ensemble du modèle.
2. **granularité de l'évolution** : les acteurs peuvent modifier le modèle de l'apprenant en tout temps.
3. **nombre de points de vue** : il y en a un par acteur intervenant dans le scénario, soit un par professeur ou tuteur et un pour l'apprenant (par défaut, il y a toujours un point de vue pour l'apprenant).
4. **stratégie de négociation** : c'est la stratégie 1 présentée au chapitre IV pages 109 et 110 qui est utilisée (si un professeur a évalué une compétence, son évaluation est conservée, si plusieurs professeurs l'ont évalué, une moyenne est faite).

5.3.2. Scénario

Deux professeurs et deux tuteurs interviennent dans ce scénario, de même qu'un apprenant, celui dont les compétences vont être évaluées. Ces acteurs sont responsables d'évaluer un certain nombre de compétences et certaines compétences sont évaluées par différents acteurs (permet de démontrer la fonction de négociation).

Trois productions sont utiles au déroulement du scénario. La première production est un tableau Excel permettant d'illustrer la compétence C2, la seconde est un fichier texte illustrant la compétence C7 et la troisième production est un enregistrement audio illustrant la compétence C8. Les descriptions détaillées de ces productions se trouvent à l'appendice I page 232.

Le scénario comporte 12 étapes combinant l'ensemble des actions effectuées par les acteurs interagissant avec le modèle dans cette simulation, comme ajouter une

compétence, illustrer une compétence par une production ou négocier une compétence. Le détail du déroulement de ce scénario se trouve à l'appendice I page 233.

5.3.3. Simulation

Dans cette section, nous précisons les résultats de la simulation effectuée dans le contexte précédent et avec le scénario présenté. Chaque étape du scénario nous a permis de valider un élément de notre système (ou le bon fonctionnement d'un cas de figure de la stratégie de négociation 1). Aussi, à chaque étape, une version du modèle de l'apprenant est créée dans SVN afin d'assurer la gestion des versions.

Au cours de cette simulation, nous avons testé et validé le bon fonctionnement des éléments suivants de Nosma :

- l'ajout d'une compétence
- le cas de figure 1 de la stratégie de négociation 1 (présentée dans le chapitre IV pages 109 et 110) : ce cas s'applique lorsqu'un seul acteur a évalué la compétence, la compétence est placée dans le point de vue consensus et supprimée du point de vue de l'acteur concerné
- l'ajout d'une production
- l'ajout d'un lien entre une production et une compétence
- le cas de figure 2 de la stratégie de négociation : plusieurs acteurs du même statut ont évalué une compétence, une moyenne est donc effectuée et les noms des acteurs ayant participé à la négociation apparaissent dans le contexte de la compétence (qui est placée dans le point de vue consensus)
- le cas de figure 3 de la stratégie de négociation 1 : la compétence est évaluée par un professeur et un tuteur et c'est l'évaluation du professeur qui est conservée dans le modèle consensus
- la certification d'une production par un acteur spécifique
- la certification d'un lien entre une compétence et une production par un acteur spécifique
- le cas de figure 4 de la stratégie de négociation 1 : c'est-à-dire que l'apprenant et un autre type d'acteur évalue la compétence; lors de la négociation, la compétence est placée dans le point de vue consensus avec

le niveau donné par le professeur (et supprimé du point de vue du professeur) et elle reste dans le point de vue de l'apprenant puisque celui-ci sera conservé même à la fermeture du modèle contexte temporaire

- le cas de figure 4 de la stratégie de négociation 1 (variante) : la compétence est évaluée par trois acteurs, deux professeurs et l'apprenant; le résultat de la négociation est que la compétence est placée dans le point de vue consensus avec la moyenne des évaluations des deux professeurs, et elle est supprimée des points de vue des professeurs mais conservée dans le point de vue de l'apprenant
- le cas de figure 5 de la stratégie de négociation 1 : les trois types d'acteurs évaluent la compétence, l'évaluation du professeur est conservée dans le modèle consensus, l'évaluation du tuteur n'est pas prise en compte et l'évaluation de l'apprenant est conservée dans son point de vue
- la fermeture du modèle contexte temporaire : à la fin du scénario d'utilisation, ce modèle temporaire disparaît puisque la formation est terminée. Toutes les compétences encore présentes dans les points de vue sont automatiquement négociées et placées dans le modèle principal.

La figure V-1 illustre cette simulation et présente le modèle tel qu'affiché à l'écran. Sur cet exemple, nous voyons particulièrement que la compétence C4 se situe dans deux points de vue différents. Notons qu'à cette étape, la fonction de négociation n'a pas été appliquée à C4 puisque celle-ci se trouve encore dans différents points de vue.

modèle de l'apprenant

PDV du tuteur Julien Contamines

Compétence C2
Description : Planifier ses apprentissages
Auteur : Julien Contamines
Habilité : Synthétiser du référentiel *Référentiel de Gilbert Paquette*, rang 8
Connaissance : Ses apprentissages
Ontologie :
Niveau : Expertise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 4.0
Contexte : cours INF1234 (catégorie 1)

Compétence C4
Description : Appliquer les techniques de communication avec le tuteur et les pairs à l'aide des outils proposés dans le cours
Auteur : Julien Contamines
Habilité : Appliquer du référentiel *Référentiel de Gilbert Paquette*, rang 5
Connaissance : Techniques de communication
Ontologie :
Niveau : Maîtrise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 3.0
Contexte : cours INF1234 (catégorie 1)

Compétence C5
Description : Auto-évaluer ses apprentissages
Auteur : Julien Contamines
Habilité : Auto-controler du référentiel *Référentiel de Gilbert Paquette*, rang 10
Connaissance : Ses apprentissages
Ontologie :
Niveau : Expertise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 4.0
Contexte : cours INF1234 (catégorie 1)

PDV de la tutrice Valéry Psyché

Compétence C4
Description : Appliquer les techniques de communication avec le tuteur et les pairs à l'aide des outils proposés dans le cours
Auteur : Valéry Psyché
Habilité : Appliquer du référentiel *Référentiel de Gilbert Paquette*, rang 5
Connaissance : Techniques de communication
Ontologie :
Niveau : Sensibilisation du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 1.0
Contexte : cours INF1234 (catégorie 1)

Ordinateur | Mode protégé : désactivé | 100%

Figure V-1. Modèle de l'apprenant simulation 1

5.4. Simulation 2 : élaboration d'un bilan de compétences

Dans cette simulation, nous nous situons dans un contexte particulier où c'est l'apprenant qui choisit de faire un bilan de compétences. Ainsi, c'est cet acteur qui pilote l'ensemble du scénario. La deuxième stratégie de négociation est utilisée, celle où l'apprenant choisit quelle évaluation il veut conserver.

Cette simulation nous permet de montrer une fois encore le bon fonctionnement de notre système mais plus particulièrement l'ajout d'une expérience professionnelle et d'un lien entre celle-ci et une compétence, la stratégie de négociation 2 et l'utilisation du modèle avec d'autres référentiels que ceux de Paquette.

5.4.1. Contexte

5.4.1.1. Description du contexte

Pour cette simulation, nous utilisons un contexte issu d'un texte publié en 2006 par le Ministère de l'Éducation, du Loisir et du Sport du Québec intitulé « Échelles des niveaux de compétence ». Ce document présente un référentiel de compétences et une échelle de niveau de compétence, s'apparentant à notre référentiel de niveaux. Dans ce contexte, les compétences sont présentées sous forme d'énoncés sans lien avec une connaissance identifiée dans une ontologie, ni avec une habileté (il n'y a donc pas de référentiel d'habiletés). Ce contexte peut donc être considéré comme ayant un référencement sémantique pauvre. Toutefois, il nous permet de montrer qu'il est possible d'utiliser notre modèle de l'apprenant sans définir les compétences selon la définition de Paquette (2002a). Il a été décidé dans ce contexte d'utiliser le champ *connaissance* de la compétence pour définir le domaine, la discipline d'application de la compétence. Ceci constitue une utilisation détournée de notre modèle mais tout à fait acceptable. Un établissement scolaire peut effectivement choisir de définir ses compétences différemment de la façon envisagée dans notre proposition. Toutefois,

cela réduit considérablement les possibilités d'interopérabilité ainsi que la richesse du référencement sémantique puisqu'il n'y a plus d'élément référencé sémantiquement.

De plus, même si ce référentiel de compétences cible l'enseignement au secondaire, dans le cadre de notre simulation, ce contexte est transposé au cas où l'apprenant fait un bilan de compétences. Dans ce contexte, des acteurs issus du monde scolaire vont évaluer les compétences de l'apprenant, mais c'est celui-ci qui aura le dernier mot au moment de la fermeture du modèle contexte temporaire puisqu'il choisira quelle évaluation sera conservée dans le modèle principal. Plus précisément, le contexte de cette simulation est le suivant : un apprenant fait un bilan de compétences à la fin de son secondaire. Ainsi, il demande à ses professeurs de compléter l'évaluation des compétences de leurs disciplines respectives et présentes dans le référentiel de compétences constituant le bilan. Il leur demande aussi, ainsi qu'à quelques camarades, d'évaluer des compétences transversales.

5.4.1.2. Référentiel de compétences

Nous avons ajouté au référentiel de compétences du MÉQ des compétences transversales divisées en deux disciplines : Communiquer et Prendre des décisions. Le référentiel de compétences ainsi obtenu contient 47 compétences réparties en six domaines et 14 disciplines. Les tableaux présentant ce référentiel de compétences se trouvent à l'appendice I pages 248 et 249.

5.4.1.3. Référentiel de niveaux

Le référentiel de niveaux utilisé dans ce contexte est différent de celui de Paquette (2002a). C'est le référentiel proposé par le Ministère de l'Éducation du Québec et qui est présenté dans le tableau V-1. Il contient cinq niveaux permettant d'évaluer les compétences.

Tableau V-1. Référentiel de niveaux du contexte 2 (MÉQ)

Niveau	Compétence	Jugement global en fin de cycle
1	Très peu développée	La compétence de l'élève est nettement en deçà des exigences
2	Peu développée	La compétence de l'élève est en deçà des exigences
3	Acceptable	La compétence de l'élève satisfait minimalement aux exigences
4	Assurée	La compétence de l'élève satisfait clairement aux exigences
5	Marquée	La compétence de l'élève dépasse les exigences

La première colonne *Niveau* est le rang, la seconde colonne *Compétence* définit le niveau, elle correspond au nom du niveau dans notre système. Par exemple, pour le niveau 1, la compétence est *très peu développée* par l'étudiant. La dernière colonne représente une description du niveau.

5.4.1.4. Contrat

Voici les valeurs des quatre points constituant un contrat :

1. **droits des acteurs** : les professeurs et tuteurs peuvent évaluer les compétences des disciplines dont ils sont responsables. De plus, les acteurs ayant reçu une invitation de l'apprenant peuvent évaluer ses compétences transversales.
2. **granularité de l'évolution** : les acteurs peuvent modifier le modèle de l'apprenant en tout temps.
3. **nombre de points de vue** : il y en a un par acteur ayant reçu une invitation de l'apprenant à évaluer ses compétences.
4. **stratégie de négociation** : c'est la stratégie 2 présentée au chapitre IV page 110 qui est utilisée (c'est-à-dire que si plusieurs acteurs ont évalué une compétence, les différentes évaluations sont listées et l'apprenant choisit celle qu'il veut conserver).

5.4.2. Scénario

Sept professeurs, quatre tuteurs et trois apprenants interviennent dans ce scénario. Ainsi 14 points de vue sont créés. Dans ce scénario, les professeurs et tuteurs évaluent les compétences correspondant à leur discipline d'enseignement.

Concernant les compétences transversales, quatre professeurs et deux apprenants participent à leur évaluation après l'invitation de l'apprenant qui effectue son bilan de compétence.

Deux productions sont utilisées dans ce scénario. La première est un fichier texte illustrant la compétence F2. La seconde est un fichier *Power Point* qui illustre la compétence AP2. Une expérience professionnelle (EP1) est utilisée au cours du scénario (les détails sur ces éléments se trouvent à l'appendice I pages 252 et 253).

Le scénario comporte 8 étapes qui permettent aux acteurs d'interagir avec le modèle (le déroulement du scénario est expliqué en détail à l'appendice I page 253).

5.4.3. Simulation

Le modèle de l'apprenant est modifié au fur et à mesure du déroulement du scénario et à chaque étape, une version du modèle est créée par SVN. Dans cette seconde simulation, nous avons tout d'abord testé et validé la possibilité d'utiliser notre modèle de l'apprenant avec d'autres référentiels que ceux de Paquette. Plus précisément, le référentiel de compétences utilisé n'est pas fondé sur la définition de compétence de Paquette et ne comporte ainsi pas de référentiel d'habiletés. Aussi, le référentiel de niveaux utilisé est celui du MÉQ. De plus, le champ *connaissance* de la compétence est utilisé pour préciser la discipline de la compétence, et le champ *ontologie* est utilisé pour préciser l'origine du référentiel de compétences. En effet, chaque institution peut choisir d'utiliser sa propre approche du concept de compétence. Par contre, comme nous l'avons déjà précisé, cela réduit et même détruit les possibilités d'interopérabilité et de référencement sémantique. D'interopérabilité puisqu'une autre institution ne pourra pas interpréter ces compétences étant donné que le modèle est utilisé de façon détournée. Comment savoir que le champ *connaissance* contient en fait la discipline ? Et de référencement sémantique puisque les compétences ne sont plus liées ni à un référentiel d'habiletés, ni à une ontologie.

Voici les autres éléments testés et validés dans cette simulation :

- l'ajout d'une expérience professionnelle
- l'ajout d'un lien entre une expérience professionnelle et une compétence
- le cas de figure 1 de la stratégie de négociation 2 (présentée dans le chapitre IV page 110): ce premier cas de figure est celui où la compétence a été évaluée par un seul acteur. Dans ce cas, à la négociation, la compétence est placée dans le modèle consensus et supprimée du point de vue de l'acteur qui l'a évaluée.
- le cas de figure 2 de la stratégie de négociation 2 : plusieurs acteurs ont évalué la compétence et le système affiche la liste des évaluations de la compétence à l'écran et l'apprenant sélectionne celle qu'il souhaite conserver dans le modèle consensus (la figure V-2 illustre cette situation, la compétence C1 a été évaluée par six acteurs et l'apprenant doit choisir quelle évaluation il veut conserver).

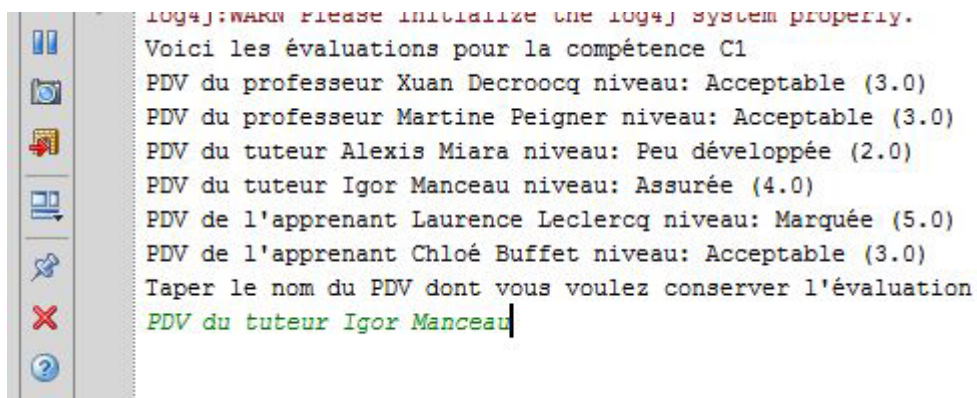


Figure V-2. Choix de l'évaluation pour C1

5.5. Simulation 3 : Système d'apprentissage en ligne

Cette troisième et dernière simulation vise à démontrer le bon fonctionnement du lien de notre système avec un système d'apprentissage en ligne, TELOS. Dans cette simulation, nous sommes dans un contexte de cours en ligne « automatique » c'est-à-dire que c'est TELOS qui ajoute des compétences dans le modèle de l'apprenant. Plus précisément, cette simulation vise à faire la preuve de l'utilisation d'ontologies et de la bonne interaction entre notre système Nosma et un autre

système, en l'occurrence TELOS; interaction qui se fera dans les deux sens, soit ajout d'information dans Nosma et demande d'information de Nosma.

5.5.1. Contexte

5.5.1.1. Description du contexte

Le contexte de cette simulation est inspiré du cours DIC 9330 *Modélisation cognitive et design pédagogique*, suivi dans le cadre du programme de doctorat en informatique cognitive. Un des travaux de ce cours consistait à construire un scénario pédagogique visant l'obtention de compétences portant sur le repérage d'information et tirées du profil de compétences du jeune chercheur nommé *Joint Skills Statement* adopté par l'*UK Research Councils* (*Research councils Uk*, 2005).

Dans cette simulation, il n'y a pas de point de vue car le scénario correspond à l'interaction de l'apprenant avec un système, TELOS. C'est ce système qui ajoute les compétences et uniquement lui, ainsi aucune négociation n'est requise. Lorsque l'apprenant termine une activité du scénario pédagogique, TELOS ajoute automatiquement une compétence dans son modèle. En réalité, une évaluation devrait probablement être faite pour s'assurer que l'apprenant possède effectivement la compétence ajoutée (que ce soit par un test automatique du type QCM ou par un acteur humain). Toutefois, dans le cadre de notre thèse et pour mener à bien notre simulation, nous considérons que TELOS peut ajouter une compétence lorsque l'apprenant termine une activité.

5.5.1.2. Référentiel de compétences

Nous avons adapté le référentiel de compétences du *UK Research Councils* pour que celui-ci soit compatible avec la définition de compétence de Paquette, c'est-à-dire que les compétences soient décrites par une connaissance (en lien avec une ontologie) et une habileté. Ce référentiel comporte huit compétences identifiées selon

un énoncé, une connaissance, une habileté (et son rang) ainsi que le niveau auquel la compétence sera évaluée au cours du scénario pédagogique. Les référentiels d'habiletés et de niveaux sont ceux de Paquette (2002a), présentés au chapitre I pages 32 et 34. Le référentiel de compétences se situe à l'appendice I page 261.

5.5.1.3. Ontologies

Pour illustrer le référencement sémantique, nous avons créé des ontologies contenant les connaissances incluses dans les compétences. Il est à noter que ces ontologies servent uniquement à illustrer le référencement sémantique en montrant comment une compétence peut être liée à une ontologie par la connaissance sur laquelle elle s'applique. Nous avons créé trois ontologies fictives (leurs représentations graphiques effectuées avec le logiciel MOT+ ONTO se trouvent à l'appendice I pages 261 à 263).

5.5.1.4. Contrat

Voici les valeurs des quatre points constituant un contrat pour ce contexte :

1. **droits des acteurs** : l'acteur dont c'est le modèle et TELOS (le système d'apprentissage en ligne) ont accès au modèle.
2. **granularité de l'évolution** : TELOS peut ajouter des informations ou questionner le modèle en tout temps. À chaque modification du modèle de l'apprenant par TELOS, une nouvelle version est créée.
3. **nombre de points de vue** : le point de vue de l'apprenant et le point de vue consensus (qui sont toujours présents).
4. **stratégie de négociation** : c'est la stratégie 3 présentée au chapitre IV page 111 qui est utilisée (c'est-à-dire que les évaluations du système TELOS sont conservées dans le modèle principal).

5.5.2. Scénario

Le scénario de la simulation correspond à un scénario pédagogique créé graphiquement avec TELOS. Ce scénario, comme nous le verrons, comporte deux « chemins ». Cela nous permet de prouver que le lien TELOS-Nosma est bien fonctionnel lors de l'interrogation de Nosma par TELOS.

Un seul acteur humain intervient dans ce scénario, l'apprenant qui effectue le scénario pédagogique.

Une production est nécessaire au bon déroulement de ce scénario. C'est un document texte intitulé *Devoir 1*.

Le scénario pédagogique a été créé graphiquement dans TELOS. Il s'intitule *Repérer les informations et les documents* (équivalent du titre du cours). La figure suivante montre le graphe de celui-ci. La flèche verte, en haut, indique le début du scénario et le carré rouge la fin de celui-ci. Les ovales sont des activités effectuées par l'apprenant ou des fonctions effectuées par TELOS et le losange représente une condition (du style si A alors B). Les valises représentent des données au format texte. La condition comporte deux cas de figure, vrai ou faux. Selon la réponse, le chemin emprunté ne sera pas le même. Il est important de noter que dans ce scénario pédagogique, une activité (la mise à niveau) n'est pas toujours effectuée. Elle le sera seulement si l'apprenant ne possède pas la compétence C7 au niveau 3 ou plus.

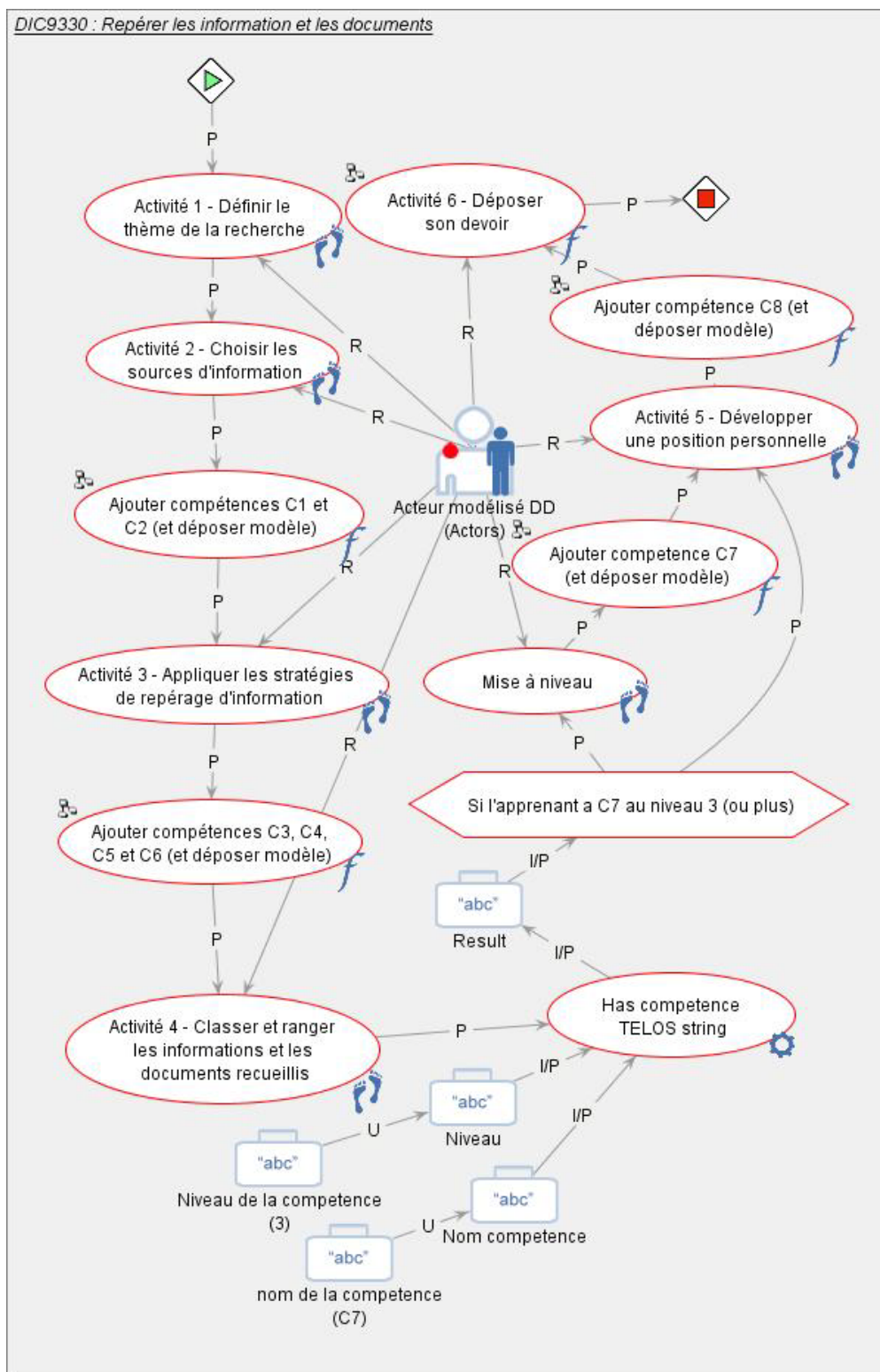


Figure V-3. Scénario pédagogique simulation 3

Le scénario (d'utilisation) comporte 11 étapes (les détails du déroulement se trouvent à l'appendice I page 266). À chaque étape, une version de modèle est créée par SVN pour assurer la gestion des versions.

5.5.3. Simulation

Le modèle de l'apprenant est modifié au fur et à mesure du déroulement du scénario. Pour tester le bon fonctionnement de la fonction d'interrogation du modèle de l'apprenant par le système d'apprentissage en ligne, nous avons effectué deux fois le scénario. Dans le cas A, le modèle de l'apprenant est vide au départ, ainsi lorsque TÉLOS interroge Nosma pour savoir si le modèle de l'apprenant contient la compétence C7 au niveau 3 ou plus, la réponse est non et l'apprenant devra ainsi effectuer la mise à niveau avant de passer à l'activité 5 (et la compétence C7 sera ajoutée au modèle). Dans le cas B, la compétence C7 se trouve dans le modèle de l'apprenant au niveau 3 dès le début du scénario, ainsi lorsque TÉLOS interroge Nosma, la réponse est oui et la mise à niveau n'est pas effectuée. Les deux cas de figure ont fonctionné comme prévu, ce qui montre le bon fonctionnement et la possibilité d'interrogation de notre système par un système d'apprentissage en ligne. Cette simulation montre aussi le bon fonctionnement de la communication entre ces deux systèmes mais dans l'autre sens, c'est-à-dire la modification du modèle par TÉLOS (en l'occurrence, l'ajout d'une compétence et l'ajout d'une production). Finalement, ce scénario montre la possibilité d'utiliser des ontologies pour le référencement sémantique des compétences.

5.6. Conclusion

Dans ce chapitre, nous avons présenté la méthode d'évaluation que nous avons choisi d'utiliser pour valider notre travail. Plus précisément, nous avons travaillé avec une approche qualitative, dans le cadre d'évaluation interne (soit en cours de

conception), nous avons effectué une *preuve d'exactitude* du prototype Nosma pour valider notre modèle conceptuel de modèle de l'apprenant. Cette validation correspond à l'évaluation du critère d'utilité de Nosma (Nielsen, 1993, Tricot *et al.* 1993). L'utilité d'un tel modèle de l'apprenant est aussi partiellement validée par les besoins du domaine identifié dans les chapitres I et II. Pour cette évaluation, nous avons effectué des simulations avec Nosma. Plus précisément, ces simulations visent à montrer la faisabilité de notre proposition en évaluant et validant le fonctionnement de notre système Nosma. Chacune des trois simulations visait l'évaluation et la validation de certaines caractéristiques de Nosma et de notre modèle de l'apprenant. Globalement, par ces simulations, nous avons validé :

- l'utilisation de points de vue;
- l'ajout d'une compétence, d'une production et d'une expérience professionnelle;
- la preuve d'une compétence par l'ajout d'un lien compétence-production ou d'un lien compétence-expérience professionnelle;
- la certification d'une production ou d'un lien;
- le fonctionnement des trois stratégies de négociation conçues et développées;
- la fermeture du modèle contexte temporaire;
- l'utilisation du modèle avec les référentiels de Paquette mais aussi avec une autre vision du concept de compétence et avec d'autres référentiels;
- l'interaction de Nosma avec un système d'apprentissage en ligne (interrogation et modification du modèle de l'apprenant).

Les simulations ont prouvé le bon fonctionnement de Nosma et ainsi la faisabilité de notre modèle conceptuel. Nous avons effectivement montré dans ce chapitre que notre proposition de modèle de l'apprenant offre des possibilités d'adaptation sémantique des apprentissages (grâce à l'utilisation de compétences), qu'elle prend en compte l'évolution dans le temps des apprentissages (avec la gestion des versions), les différents acteurs prenant part au processus d'apprentissage (avec les points de vue multiples) ainsi que les différents contextes d'occurrence des

situations d'apprentissage. Ceci constitue une première étape indispensable à tout processus de validation. L'étape suivante consiste à évaluer l'utilisabilité de Nosma (Nielsen, 2003 et Tricot et *al.*, 2003) selon les critères présentés dans la section 5.1.3. de ce chapitre.

CONCLUSION

Dans cette conclusion, nous allons rappeler dans quel domaine se situe cette thèse, ce qu'elle y apporte et ce que notre proposition a d'original. Nous allons aussi présenter les principaux résultats de l'évaluation et les pistes de recherche que nous envisageons d'explorer à l'avenir.

Rappel du contexte : la réalité de la recherche en EIAH

La problématique de cette thèse concerne la création d'un modèle de l'apprenant dans le contexte actuel de la recherche en EIAH (environnements informatiques d'apprentissage humain). Rappelons ce contexte. Les recherches en EIAH ne peuvent aujourd'hui ignorer l'émergence du web sémantique qui apporte une dimension sémantique et sociale aux EIAH qui ne peut être passée sous silence. Les pratiques en éducation sont aussi en mouvement et on dénote aujourd'hui l'utilisation, entre autres, de l'approche par compétences. N'oublions pas les standards qui constituent un aspect important des efforts d'interopérabilité des systèmes. Ces aspects influencent les recherches en EIAH et dictent certaines caractéristiques que les EIAH d'aujourd'hui doivent posséder ou visent à posséder. Ainsi, les EIAH d'aujourd'hui sont (ou visent à être) distribués, ouverts, interopérables, développés avec une vision d'apprentissage tout au long de la vie, ancrés sémantiquement et multi-acteurs.

Rappel des objectifs de recherche

Dans ce contexte, notre objectif de recherche est de proposer un modèle cognitif de l'apprenant en vue de supporter l'évolution de ses apprentissages ayant lieu dans différents contextes, avec différents acteurs et tout au long de sa vie. Un

deuxième objectif est de formaliser et d'opérationnaliser ce modèle conceptuel par un prototype que nous avons nommé Nosma.

Rappel de la démarche de recherche et des résultats obtenus

Pour atteindre notre objectif de recherche, nous adoptons le processus unifié (Larman, 2005). Nous l'avons adaptée à une démarche méthodologique de recherche en nous inspirant de méthodes de recherche-développement et de techniques de modélisation et de simulation. Finalement, notre démarche de recherche comporte trois phases : 1) l'inception, 2) l'élaboration et la construction et finalement, 3) la transition. Il faut retenir que cette démarche nous a permis de concevoir un modèle conceptuel d'un modèle de l'apprenant puis de le formaliser et de l'opérationnaliser dans un prototype, Nosma, de façon itérative.

Originalité et apport de notre contribution à la recherche en EIAH

Cette thèse a contribué à l'avancement des connaissances sur la recherche en EIAH principalement par sa proposition d'un modèle de l'apprenant. En effet, le modèle conceptuel que nous proposons contribue de façon originale au domaine des EIAH de par ses aspects qui couvrent les principales caractéristiques visées par les EIAH d'aujourd'hui. Plus précisément, notre modèle de l'apprenant est :

- un modèle cognitif de l'apprenant puisque conçu sur ses compétences
- un modèle permettant un référencement sémantique illustré puisque les compétences sont décrites en lien avec des référentiels d'habiletés et de niveaux, en lien avec des ontologies de domaines et qu'elles peuvent être illustrées, prouvées par des productions ou des expériences professionnelles de l'apprenant
- un modèle adoptant une vision tout au long de la vie des apprentissages puisqu'évoluant dans le temps grâce au système de gestion de versions intégré à Nosma, grâce à l'utilisation de la notion de contrat qui permet de « brancher » le modèle à différents systèmes et par la proposition de l'utilisation de modèles contexte et de modèles contexte temporaires

- un modèle multi-acteurs prenant en compte et intégrant les différents acteurs participant au processus d'apprentissage, grâce à la gestion de points de vue multiples
- un modèle facilitant l'interopérabilité puisque défini et stocké en XML.

Ces caractéristiques se retrouvent au cœur des recherches actuelles dans le domaine. L'originalité de cette thèse est de les regrouper dans un seul et même modèle.

Sur le plan informatique, le produit de cette thèse est Nosma, prototype ayant permis l'opérationnalisation de notre modèle conceptuel. Nosma nous a ainsi permis de faire la preuve de la faisabilité technique d'un tel modèle et de sa viabilité (tout du moins à une échelle prototypaire) par la conduite de trois simulations issues de trois contextes d'apprentissage différents auxquels ont été associés trois scénarios d'utilisation.

Toujours sur le plan informatique, mais concernant l'apport de cette thèse à la communauté scientifique, plusieurs points sont à souligner, particulièrement notre proposition originale et innovante de formalisation de la gestion simultanée de points de vue, de versions et de différents contextes. Effectivement, notre modèle de l'apprenant comporte, en cours d'utilisation, autant de versions que de contextes auxquels il est lié qui eux-mêmes comportent autant de points de vue que d'acteurs interagissant avec le modèle, et sur lesquels une gestion des versions est effectuée. Finalement, nous avons des versions de modèles d'apprenant selon trois aspects : 1) le temps (gestion des versions), 2) les acteurs (gestion de points de vue) et 3) les contextes (gestion de différents modèles contextes temporaires et d'un modèle principal reflétant la « mémoire » des contextes passés). Pour parvenir à gérer toutes ces versions de nos modèles d'apprenant, nous avons eu l'idée innovante d'utiliser une structure en trois couches (métamodèle, modèle et instance) combinée à l'idée de modèle contexte temporaires, de modèle principal et à l'existence de différents points de vue pour les acteurs de chaque contextes en cours d'utilisation.

Cette recherche nous a apporté son lot de défis, aussi bien sur l'aspect cognitif que sur l'aspect informatique de cette thèse. Principalement, sur le plan cognitif, le défi a été de concevoir un modèle de l'apprenant répondant aux besoins actuels des EIAH, soit un modèle cognitif, supportant l'apprentissage tout au long de la vie, son évolution et les différents acteurs y participant. Sur le plan informatique, l'intégration d'un système de gestion de versions existant dans Nosma, notre réflexion sur l'utilisation de contrat pour gérer les interactions de Nosma avec ses utilisateurs humains et machine ont constitué des défis que nous sommes contents d'avoir relevés. Mais notre principal défi (qui constitue notre principal apport) a été de combiner ces apports innovants issus aussi bien de l'aspect cognitif que de l'aspect informatique de cette recherche.

Le principal impact de cette recherche est, selon nous, l'adoption d'une vision globale intégrant les pratiques, conceptions et théories de différents domaines pour la conception de l'apprentissage (c'est-à-dire aussi bien des sciences de l'éducation, du génie cognitif, de la gestion des connaissances, du génie informatique...). D'autres projets de recherche sont abordés de façon multidisciplinaire et nous pensons que c'est l'avenir pour la réussite et le succès des recherches futures. L'approche que nous avons adoptée pour la conception d'un modèle d'apprenant pourrait être étendue et utilisée pour concevoir des modèles d'utilisateur dans d'autres domaines que l'éducation (la recherche sur internet, l'aide en ligne, etc.).

Diffusion de la recherche

Tout au long de la thèse, le travail présenté ici a été l'objet de différentes communications et publications que nous classons et listons ici.

Acte de conférence

Moulet, L., Marino, O., Hotte, R. et Labat, J-M. (2008). *Framework for a Competency-Driven, Multi-Viewpoint, and Evolving Learner Model*. ITS 2008, Montréal. 702-705.

Moulet, L., Marino, O., Hotte, R. et Labat, J-M. (2008). *Méta-modèle pour la création de Modèles de l'Apprenant Cognitifs, à Points de Vue Multiples et Évolutifs*. TICE 2008, Paris.

Moulet, L. (2006). *ePortfolios, compétences et personnalisation des apprentissages*. Colloque CIRTA au congrès international de l'ACÈD et de l'AMTEC. 23-26 mai 2006. Montréal, Canada.

Moulet, L., Marino, O., & Hotte, R. (2006). *Holistic evolving and multi-viewpoints learner model*. Actes de la conference I²LOR-06, Third annual scientific conference - LORNET Research Network. Montréal Canada.

Note de recherche

Moulet, L. (2005). Revue de littérature du ePortfolio: Définitions, contenus et usages. Visant à l'intégration d'un ePortfolio dans le modèle de l'apprenant d'un système d'apprentissage en ligne. Note de recherche LICEF06NR02, Montréal, LICEF, Télé-université.

Communication orale:

Moulet, L. (2005). ePortfolio, competencies and learner model. ePortfolio Pan American working forum. Vancouver, Canada. 18, 19 avril 2005.

Communication affichée:

Moulet, L. (2005). ePortfolio as learner models ? I2LOR-05, Second annual scientific conference - LORNET Research Network. 16-18 novembre 2005. Vancouver, Canada.

Moulet, L. (2004). Actor modeling: trends and challenges. Competency oriented learner model. I²LOR-04, First annual scientific conference - LORNET Research Network. Montréal, Canada.

Perspectives et pistes de recherche

Nous identifions des pistes de recherche sur trois plans, sur le plan informatique, sur le plan conceptuel et au niveau de l'évaluation de notre recherche.

Sur le plan informatique : une première perspective concerne la prise en compte **des normes et standards**. Cet aspect est même essentiel lorsque l'on se situe dans une vision d'apprentissage tout au long de la vie. Ces normes et standards peuvent concerner différents éléments, comme les compétences, le modèle de

l'apprenant dans son ensemble, les contrats... Pour la perrénité du projet, il est très important de suivre l'évolution de la standardisation dans le domaine et d'ajuster notre modèle de l'apprenant pour répondre aux exigences des normes et standards utilisés par la communauté scientifique. De plus, cet aspect touche la **persistance du modèle** dans le temps (encore une fois, point essentiel pour l'apprentissage tout au long de la vie), d'une part des modèles d'apprenant existants mais aussi du modèle conceptuel qui sera amené à évoluer (pour répondre aux standards, pour s'améliorer, pour être enrichi, etc.).

Sur le plan conceptuel : une perspective intéressante concerne les **compétences**. Une compétence est-elle persistante ? Une compétence acquise l'est-elle pour toute la vie ? Ou bien au contraire, est-ce que les compétences « se perdent » avec le temps ? Ces questions sont très importantes et méritent d'être approfondies conceptuellement dans un premier temps puis mises en place informatiquement si nécessaire. Dans le sens que si des règles sur la persistance dans le temps sont écrites, alors il faudra développer un moyen informatique pour les valider dans les modèles d'apprenant. Aussi, les liens entre une compétence et une production ou une expérience professionnelle ne sont pas validés si la compétence est modifiée. La question est si telle production illustre telle compétence au niveau 2, alors l'illustre-t-elle aussi au niveau 4 ? Une réponse rapide serait : oui mais elle ne l'illustrera peut-être pas aussi justement et précisément. Mais une réponse complète nécessiterait plus de recherche. Cet aspect de la persistance et de la validité des liens selon l'évolution des compétences n'a pas été envisagé dans Nosma. Il serait très souhaitable de l'étudier dans la suite de cette recherche. Une autre piste de recherche très intéressante concerne la réflexion autour de l'utilisation du modèle par l'apprenant pour l'aider à autogérer ses compétences et pour l'aider à effectuer un **retour réflexif sur ses apprentissages**. Cette piste constitue une nouvelle avenue de recherche très large qui nécessite une étude approfondie de la documentation scientifique sur le sujet. Élargir l'utilisation possible de Nosma en y ajoutant des fonctionnalités sur la base de cette étude de la documentation soulèvera à coup sûr de

nouvelles problématiques dont la résolution représente un projet de recherche à part entière.

Sur le plan de l'évaluation : une piste de recherche évidente est d'effectuer la deuxième phase d'évaluation. Nous avons effectué dans le cadre de cette thèse une première étape d'évaluation et de validation qu'il serait très intéressant et opportun de compléter par une seconde phase qui impliquerait des acteurs humains. Comme nous l'avons vu lors de l'étude des méthodes d'évaluation, le critère utilité (que nous avons évalué) est complété par le critère d'utilisabilité. L'implication d'acteurs humains servirait l'évaluation des deux critères, d'une part, avec la consultation d'experts et de l'autre avec la consultation de tous les types d'utilisateurs finaux du modèle (professeur, étudiant, etc.). La consultation d'expert permettrait de renforcer la validation de la pertinence de notre modèle conceptuel et offrirait très certainement des résultats pouvant nourrir une nouvelle boucle dans notre démarche (c'est-à-dire des idées d'améliorations de notre recherche que nous pourrions prendre en compte pour modifier notre modèle conceptuel puis son implémentation dans Nosma). L'implication des utilisateurs finaux permettra d'effectuer des tests d'utilisabilité, essentiel en développement de logiciel. Rappelons que notre projet est itératif et que l'évaluation permet de détecter des problèmes ou de nouvelles idées qui sont sources pour le début d'une autre itération. Nous sommes arrivée à la fin de la boucle, soit la phase 3 de notre démarche de recherche et toute évaluation effectuée à ce stade nourrira une nouvelle boucle (ou itération), ce qui constitue une piste de recherche très intéressante qui sera examinée dans une recherche future.

APPENDICE. A - LÉGENDE DU LOGICIEL MOTPLUS ONTO

MOTPlus ONTO est un logiciel qui permet de représenter graphiquement des ontologies et de les exporter en OWL.

Cet appendice est tiré du manuel de l'utilisateur de MOTPlus ONTO (version 1.6.5), rédigé en 2007, par Jacques Rivard, avec la collaboration de Claire Banville, Martin Deveault, Denis Gareau, Michel Léonard, Stefan Mihaila, Gilbert Paquette et Ioan Rosca.

A.1 Type d'objet du modèle ontologique

MOTPlus permet de créer des objets graphiques représentant des types d'objets abstraits (Concept et principe), des types d'objets concrets (faits), des types de relations entre les objets précédents. Dans le modèle de type ontologique, les objets sont classés en trois catégories (Concept, Principe, Fait), et les éléments OWL-DL sont représentés chacun par un sous-type d'objet dans chacune de ces catégories (voir la Figure A-).

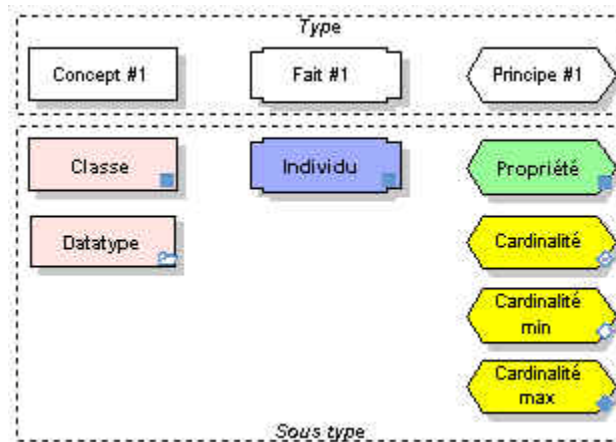
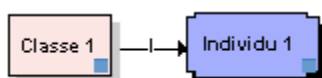


Figure A-1. Sous type d'élément OWL

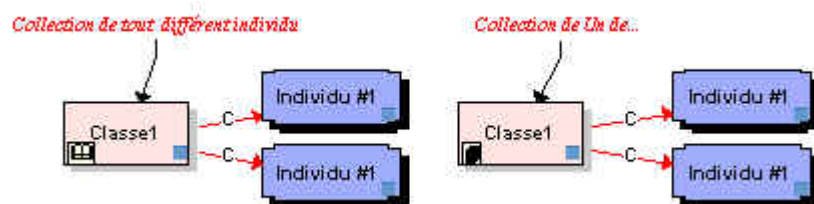
A.2 Description des liens du modèle ontologique utilisés

Dans le type de modèle Ontologie de MotPlus, les types d'objets peuvent être liés entre eux par plusieurs types de liens. Cependant, nous présentons ici seulement les que nous avons utilisés dans le document présent, l'interprétation de l'ensemble de liens disponibles étant accessible à partir du guide d'utilisation de MOTPlus ONTO.

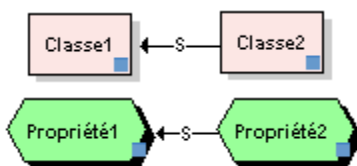
- **Le lien d'Instanciation (I)** relie une classe à un Individu (Instance de la classe).



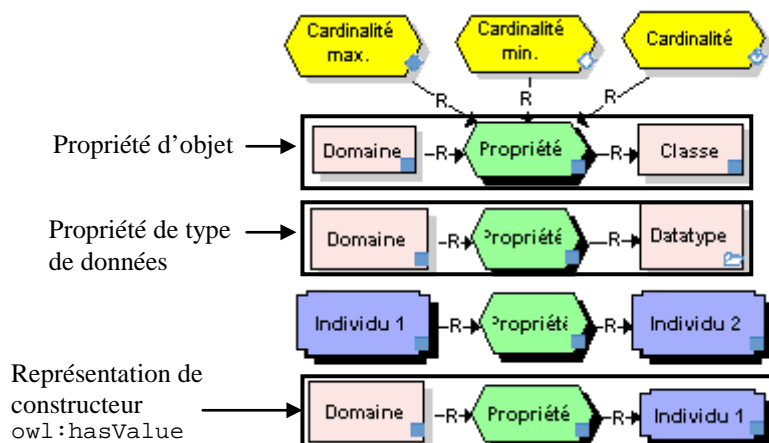
- **Le lien de Composition (C)** relie une Classe à une collection d'Individu.



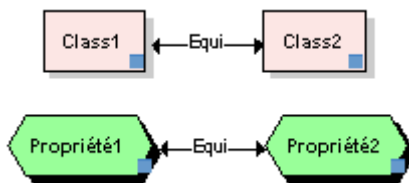
- **Le lien de Spécialisation (S)** met en relation deux Classes ou deux Propriétés dont l'une est « une sous-classe » ou « sous-propriété » de l'autre.



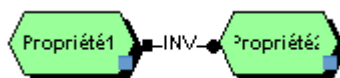
- **Le lien de Régulation (R)** met en relation, dans un sens ou dans l'autre, une propriété et, une classe, un datatype ou un individu, afin d'exprimer le domaine et le codomaine de la propriété. Le lien (R) met aussi en relation les cardinalités et les propriétés, afin de créer des restrictions.



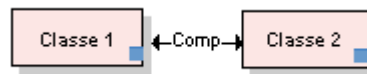
- **Lien Équivalent (Equi)** permet de relier une description de classe à une autre description de classe pour déclarer leur équivalence. Toutefois les classes équivalentes ne sont pas égales, c'est-à-dire n'ont pas la même signification intensionnelle (ne représentent pas le même concept). Ce lien déclare aussi que deux propriétés sont équivalentes.



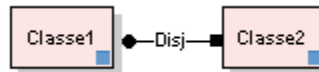
- **Lien Inverse (Inv)** déclare une propriété comme étant l'inverse d'une autre.



- **Lien Complément de (Comp)** relie une classe à son complément (c.-à.-d. la classe dont l'extension contient exactement les individus qui n'appartiennent pas à l'extension de la description de classe faisant l'objet de la déclaration).



- **Lien Disjointes (Disj)** relie deux descriptions de classe disjointes, qui n'ont aucun individu en commun.



APPENDICE. B - ANALYSE DE DÉFINITIONS D'EPORTFOLIO

Analyse des définitions des ePortfolios selon les critères Contenu et Usage

Définitions	Contenu		Usage	
	Structure	Sémantique	Services	Buts
OSPI			<i>saving, organizing, viewing, and selectively sharing</i>	
ePortConsortium			<i>organize and present work</i>	<i>a context for discussion, review and feedback, demonstrate progress and accomplishments</i>
Educause	<i>a collection of authentic and diverse evidence</i>	<i>drawn from a larger archive representing what a person has learned over time, on which the person has reflected</i>		<i>for presentation to one or more audiences for a particular rhetorical purpose</i>
Eifel	<i>collection d'informations numériques</i>	<i>décrivant et illustrant l'apprentissage ou la carrière d'une personne, son expérience et ses réussites</i>		<i>validation des acquis de l'expérience, compléter ou remplacer des examens, réfléchir sur son apprentissage ou sa carrière, accompagner le développement professionnel continu, la planification de l'apprentissage ou la recherche d'un travail.</i>
ERADC	<i>a digital repository of artefacts</i>		<i>Accreditation for prior and/or extra-curricular experiences, control over access</i>	<i>demonstrate competence and reflect on their learning, achieve a greater understanding of their individual growth, learning and career planning</i>
LIFIA	<i>personal profile and collection of achievements</i>	<i>describing and documenting a person's achievements and learning</i>		<i>accreditation of prior experience, job search, continuing professional development, certification of competences</i>

Analyse des définitions des ePortfolios selon les critères Contenu et Usage (suite)

Définitions	Contenu		Usage	
	Structure	Sémantique	Services	Buts
ePortfolio Portal	<i>a digital repository of artifacts</i>		<i>Accreditation for prior and/or extra-curricular experiences, control over access</i>	<i>to demonstrate competence and reflect on their learning, achieve a greater understanding of their individual growth, career planning and CV building</i>
IMS	<i>Collections d'informations appartenant à l'apprenant</i>	<i>rassemblant les résultats de ses études ou de ses formations, ses buts, ses expériences professionnelles</i>		<i>des bulletins de notes, la planification d'un plan de carrière et la possibilité de postuler à un emploi, ou définir un apprentissage personnalisé.</i>
SPARC				<i>to track, reflect on, and plan their learning objectives and growth, to combat plagiarism and engineer personal pride in school-work, help them accomplish their career or academic goals.</i>
Penn State Univ.	<i>selected evidence from coursework, artifacts from extra-curricular activities</i>	<i>reflective annotations and commentary related to these experiences</i>		<i>plan, reflect upon, and publish</i>

APPENDICE. C - ANALYSE D'OSP

C.1 Qu'est-ce qu'OSP

OSP est un outil supportant des activités liées à un portfolio. Il fournit un environnement sur lequel une personne (le propriétaire de l'*ePortfolio*) peut montrer son travail. L'environnement propose des outils pour aider l'utilisateur à sélectionner des preuves présentant le mieux ses accomplissements, ses apprentissages, ses buts, à mieux y réfléchir et à mieux les présenter à une tierce personne (un collègue, un professeur, un futur employeur...). L'organisation de l'outil tourne autour de *Common Interest Groups* (CIG). Les utilisateurs adhèrent à ces groupes pour échanger/présenter leur travaux personnels. Un tel groupe pourrait par exemple être constitué par les étudiants suivant un même cours à une même session. Chaque *Common Interest Groups* a un utilisateur coordinateur qui joue le rôle d'administrateur pour le CIG concerné.

L'environnement fournit aussi des outils spécifiques pour ces coordinateurs de CIG, ainsi que pour les évaluateurs et les administrateurs. Ces outils leur permettent de guider les utilisateurs d'OSP (nous verrons plus en détail ce que ces outils permettent de faire dans le paragraphe concernant les fonctionnalités d'OSP). Deux types d'utilisateurs existent donc pour OSP et chacun de ces deux types bénéficie de fonctionnalités différentes; d'un côté, les propriétaires d'*ePortfolio* et de l'autre les administrateurs, les coordonnateurs de groupes et les évaluateurs.

OSP est une application Java qui tourne sur un serveur et est accessible aux utilisateurs par un navigateur internet (comme *Internet Explorer* ou *Mozilla Firefox*). Cet outil fait partie du réseau Sakai. Il peut être utilisé sans connaissance informatique en utilisant la documentation créée pour installer et utiliser l'outil. En

revanche, pour le modifier, il faut être familier avec les technologies XML, XSL, XSLT, XHTML, JSP, SQL, l'architecture *Sakai*, *Spring* et *Hibernate*.

En réalité, il n'est pas aisé d'utiliser OSP sans avoir de compétences informatiques. Un *quick start guide* est disponible pour l'installation³ mais des erreurs surviennent lors de celle-ci et il existe en réalité peu de documentation pour comprendre le fonctionnement de l'outil, pour connaître les différentes fonctionnalités qu'il propose et pour les utiliser. Les outils de communication efficaces pour obtenir de l'aide sont les forums de discussion sur lesquels on peut poser des questions qui obtiennent des réponses rapides (dans la journée), des personnes participant au projet sont ainsi prêtes à aider les nouveaux venus. Toutefois, les forums de discussion sont surtout dédiés aux difficultés techniques rencontrées par des programmeurs installant ou utilisant OSP dans leur institution. Nous allons maintenant présenter quelles sont les fonctionnalités d'OSP.

C.2 Principales fonctionnalités d'OSP

Les utilisateurs accèdent à OSP par l'entremise d'un navigateur internet. Une fois qu'ils sont connectés, ils accèdent à leur page d'accueil. Celle-ci contient en haut une liste d'onglets correspondant au CIG auxquels l'utilisateur est inscrit et pour chacun de ces CIG un menu d'outils est disponible à gauche de l'interface. À titre d'exemple, la figure C-1 présente la page d'accueil d'un utilisateur administrateur.

Tous les utilisateurs ont un onglet intitulé *MyWorkplace*. Sur cet espace, un administrateur peut créer de nouveaux CIG alors qu'un utilisateur peut uniquement adhérer ou se désabonner à des CIG. L'administrateur dispose aussi d'un onglet *Administration Workspace* qui lui permet de gérer les utilisateurs. Passons maintenant en revue l'ensemble des fonctionnalités d'OSP.

³ <http://www.theospi.org/>

Home : cet outil est la page d'accueil de l'onglet. Généralement, il contient des messages d'information concernant le CIG représenté par l'onglet.

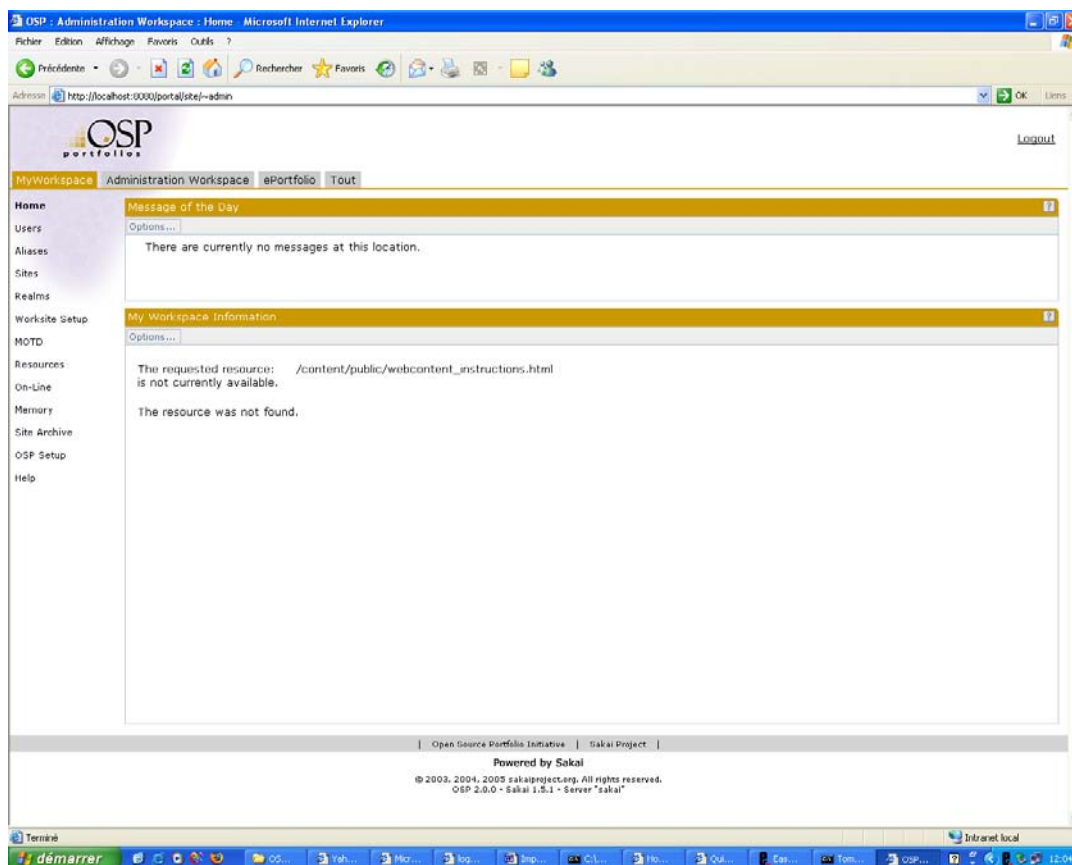


Figure C-1. Page d'accueil d'un administrateur

Schedule : cet outil permet de visualiser un emploi du temps (ou de le créer si l'utilisateur est administrateur pour le CIG concerné). L'emploi du temps peut être affiché par jour, semaine, mois, an ou par liste d'évènements. Les activités inscrites dans l'emploi du temps peuvent être de différents types (Academic Calendar, Class Session, Class Session – Discussion, Class Session – Small Group, Deadline, Class Session – Lecture, Class Session – Lab, Multidisciplinary Conference, Web Assignment, Activity, Exam, Quiz, Cancelled, Computer Session, Meeting, Special Event). Pour créer une nouvelle activité, l'utilisateur doit remplir un formulaire, contenant le titre de l'activité, son type, sa durée, sa fréquence, son lieu et autres. On

peut aussi importer un calendrier existant et générer le calendrier au format PDF pour pouvoir l'imprimer. La Figure C- montre un emploi du temps présenté par semaine.

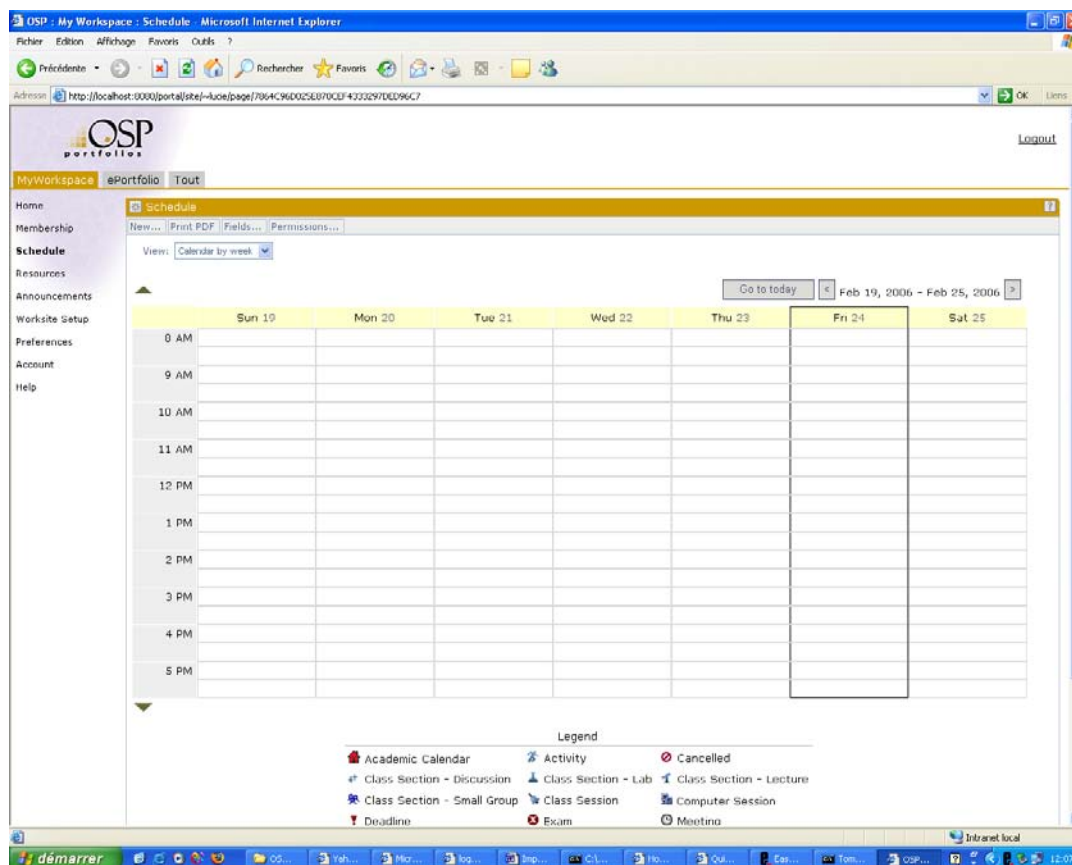


Figure C-2. Emploi du temps par semaine

Announcements : cet outil permet de créer des annonces et de les visualiser. Ces annonces peuvent ensuite être sélectionnées pour être présentées sur des pages d'accueil (par exemple sur la page *Home*).

Ressources : cet outil permet à l'administrateur de gérer un espace de ressources et aux utilisateurs d'accéder aux ressources mises à leur disposition.

Discussion : cet outil est un forum de discussion dans lequel l'administrateur peut créer des catégories et des sujets dans ces catégories. L'utilisateur peut uniquement répondre à un sujet posté par un administrateur.

Assignments : cet outil permet à l'administrateur de créer une boîte aux lettres pour réceptionner des travaux d'étudiants. Les utilisateurs peuvent sélectionner un

assignments et lire les instructions qui y sont associées, téléverser un document (leur travail à rendre) etc. L'outil offre aussi un système de notation (en lettre, chiffre).

Drop Box : cet outil permet au professeur et aux étudiants de partager des documents. Il y a un dossier pour chaque étudiant (privé).

Chat Room : cet outil permet de faire du clavardage aux participants du CIG.

Email archive : chaque site correspondant à un CIG dispose d'une adresse de messagerie générée automatiquement, cet outil permet de voir les courriels envoyés à la liste (constituée des membres du CIG).

Sakai news : cet outil présente les informations diffusées sur le site de Sakai (les nouvelles du projet).

Web Content : cet outil permet de visualiser dans l'interface d'OSP le contenu d'un site externe.

Repository : cet outil permet aux utilisateurs de gérer un espace de stockage (voir figure C-3 page suivante).

Glossary : cet outil permet à l'administrateur de créer un glossaire (lié au CIG).

Les trois outils restant à présenter sont liés, ils concernent la création de présentations. Les deux premiers concernent la création de présentations à partir de gabarits ainsi que la création de gabarit. Le troisième outil nécessaire à la création de présentation se nomme *Forms*, il permet de créer des formulaires. Une présentation permet à un étudiant de montrer des informations le concernant (ses apprentissages, ses réussites, ses expériences professionnelles...) à une tierce personne. Comme nous le verrons, l'étudiant peut choisir les personnes auxquelles il va donner accès aux présentations qu'il crée.

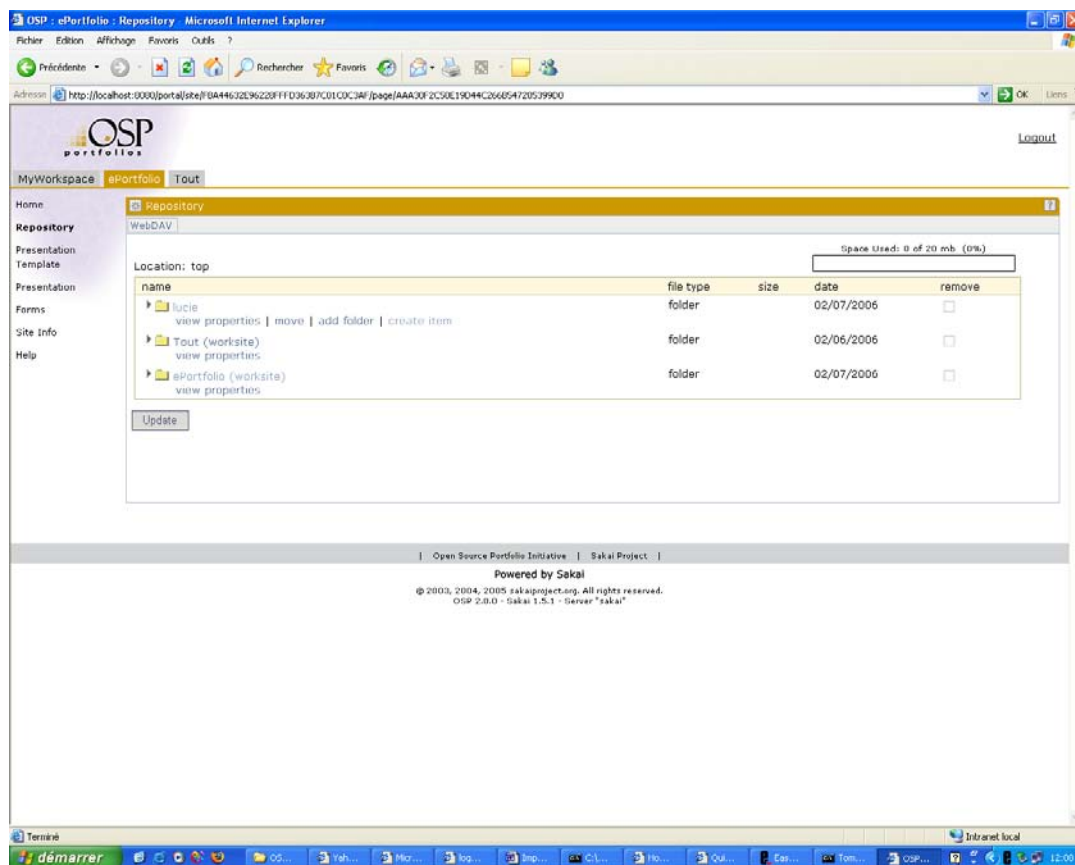


Figure C-3. Espace de stockage

Pour qu'un utilisateur puisse créer une présentation, il faut que l'administrateur ait créé (ou importé) des gabarits de présentation. Pour créer des gabarits, l'administrateur doit tout d'abord créer (ou importer) des formulaires. Un gabarit de présentation est donc un ensemble de formulaires. Un utilisateur doit ensuite instancier les formulaires avant de créer la présentation pour ensuite pouvoir compléter la présentation avec les formulaires instanciés déjà existants. Détaillons cette tâche. Tout d'abord, l'utilisateur appuie sur le bouton *Nouvelle présentation*, apparaît alors la page de création de présentation. Il doit alors préciser le titre de la présentation, en faire éventuellement une description et sélectionner un gabarit à partir duquel la présentation va être générée. Ensuite, l'utilisateur doit sélectionner les formulaires instanciés qu'il veut voir apparaître dans la présentation. L'utilisateur ne peut pas instancier de formulaires à cette étape, au moment où il crée la présentation,

les formulaires doivent déjà être remplis. Ceci exige que l'utilisateur doive savoir quels formulaires existent dans les gabarits de présentation avant de créer une présentation à partir d'un de ces gabarits. La troisième étape de création de présentation consiste à spécifier les droits des autres utilisateurs sur la présentation (lecture, lecture et écriture...) et la dernière étape permet de prévenir certaines personnes de la création de la présentation (par exemple pour obtenir l'avis de ces personnes sur la présentation, ou pour inviter un futur employeur à aller visualiser notre CV).

APPENDICE. D - ANALYSE DES SYSTÈMES DE GESTION DE VERSIONS

Les 37 critères d'analyse des outils de gestion de versions et les tableaux comparatifs des 37 outils

Tableau D-1 Critères d'analyse des systèmes de gestion de versions – Catégorie Information générale

Critère	Description
Statut de développement	Décrit le statut courant de développement du logiciel.
Modèle du répertoire	Décrit la relation entre les différentes copies présentes dans le répertoire. Deux types de modèle: client-serveur (centralisé) et distribué (décentralisé).
Modèle de l'accès simultané	Décrit comment les changements sont gérés afin d'éviter les inconsistances. Deux types de modèle: <i>lock</i> et <i>merge</i> .
Licence	Spécifie la licence du logiciel.
Licence libre	Spécifie si le logiciel est sous licence libre ou non.
Portabilité	Décrit si le logiciel est utilisable sur différents systèmes d'exploitation, architectures et autres types de systèmes.
Plateformes supportées	Spécifie les systèmes d'exploitation supportés par le logiciel.

**Tableau D-2 Critères d'analyse des systèmes de gestion de versions – Catégorie
Information technique**

Critère	Description
Langage de programmation	Spécifie le langage de programmation utilisé pour le développement de l'application.
Modèle de l'historique	Décrit la forme sous laquelle les changements sont stockés dans le répertoire. Deux types: <i>snapshot</i> (conservation d'une copie du répertoire avant et après le changement), <i>changeset</i> (conservation d'une copie avant le changement et d'un ensemble de changements).
ID des revisions	Décrit comment les différentes versions sont identifiées à l'interne du logiciel. Exemples de type: identifiant random, tables, nom de fichier séquentiel (<i>namespace</i>).
Taille du repertoire	Décrit comment le répertoire grandit au fur et à mesure des changements. Deux types: <i>O(patch)</i> grossit de la taille des patch entre les révisions, <i>O(revisions)</i> grossit de la taille de chaque révision enregistrée.
Documentation disponible	Décrit la qualité de la documentation disponible sur le logiciel.
Facilité de déploiement	Décrit la facilité avec laquelle on peut déployer le logiciel.
Ensemble de commandes	Décrit l'ensemble de commandes du logiciel et sa compatibilité avec CVS.
Intégration du réseautage	Spécifie le degré d'intégration du réseautage dans le système ainsi que la conformité aux protocoles existants.
Protocoles réseau	Spécifie les protocoles utilisés pour la synchronisation des changements.

Tableau D-3 Critères d'analyse des systèmes de gestion de versions – Catégorie Fonctionnalités

Critère	Description
Validation atomique	Garanti que tout changement fait sera fusionné, sinon pas de changement. Garanti que si un changement est interrompu, le répertoire ne sera pas inconsistant.
Renommage de fichier	Spécifie s'il est possible de renommer un fichier tout en gardant son historique.
Copie de fichier ou de dossier	Spécifie s'il est possible de copier un fichier à un autre emplacement dans le répertoire tout en gardant son historique.
Copie du répertoire	Spécifie s'il est possible de copier le répertoire afin d'obtenir une copie fonctionnelle sur le système local.
Propager les changements	Spécifie si le système peut propager les changements aux répertoires parents.
Définir des permissions	Spécifie s'il est possible de définir des permissions d'accès pour des parties du répertoire.
Ensemble de changements	Spécifie si le répertoire supporte les ensembles de changements (façon de grouper un ensemble de modifications liées en un élément atomique).
Historique du fichier ligne par ligne	Spécifie si le système de gestion de versions a une option permettant de suivre l'historique du fichier ligne par ligne.
Possibilité de travailler sur un seul dossier du répertoire	Spécifie si le système de gestion de versions peut travailler sur un seul dossier du répertoire.
Possibilité de traquer les changements courants	Spécifie s'il est possible de suivre les changements courants qui n'ont pas encore été validés dans le répertoire.
Fusion de fichiers renommés	Décrit si le système peut fusionner les changements faits sur un fichier d'une branche au même fichier qui a été renommé sur une autre branche.
Liens symboliques	Décrit si le système permet la gestion de versions de liens symboliques.
Ancrage pré et post événement	Indique la capacité de déclencher des commandes avant ou après une action.
Signature des révisions	Fait référence à l'intégration d'une signature digitale aux changements.
Traçabilité des fusions	Décrit si le système se rappelle des changements qui ont été fusionnés entre différentes branches pour ne fusionner que les changements manquants.
Caractère de fin de ligne	Décrit si le système peut adapter le caractère de fin de ligne aux fichiers textes pour qu'ils correspondent au style de fin de ligne du système d'exploitation utilisé.
Tags	Précise s'il est possible de donner des noms aux changements.
Support international	Précise si le logiciel a un support multi-langues.

**Tableau D-4 Critères d'analyse des systèmes de gestion de versions – Catégorie
Interface utilisateur**

Critère	Description
Interfaces web	Spécifie si le système dispose d'une interface Web.
Interfaces graphiques	Spécifie si le système dispose d'une interface graphique.
Plug-ins pour IDE	Disponibilité de plug-ins pour des IDE (<i>Integrated Development Environment</i>).

Nous présentons maintenant les trois tableaux comparant les 16 systèmes de gestion de versions sous licence libre. Les tableaux suivants présentent la comparaison de l'ensemble des systèmes de gestion de versions répertoriés au cours de notre recherche.

Tableau D-5 Comparatif des 16 outils de gestion de versions sous licence libre (1)

Catégorie	Critère	Aegis	Arch	Bazaar	Codeville	CVS
Information générale	Statut de développement	?	maintenu mais pas de nouveau développement	développement actif	développement actif	maintenu mais pas de nouveau développement
	Modèle du répertoire	?	distribué	distribué	distribué	client serveur
	Modèle accès simultané	?	merge	merge	merge	merge
	Licence	GPL	GPL	GPL	BSD	GPL
	Licence libre	oui	oui	oui	oui	oui
	Portabilité	moyen	bon	excellent	?	bon
	Plateformes supportées		Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	?	C, shell scripts	Python	?	C
	Documentation disponible	moyen	moyen	excellent	?	excellent
	Facilité de déploiement	moyen	excellent	très facile	?	bon
Fonctionnalités	Validation atomique	oui	oui	oui	?	non
	Définir des permissions	oui	oui	prévu	?	limité
	Ensemble de changements	oui	oui	oui	?	non
	Historique du fichier ligne par ligne	oui	non	oui	?	oui
	Signature des révisions	?	oui	partiel	?	non
	Tags	?	oui	oui	?	oui
	Support international	?	?	oui	?	?

Tableau D-6 Comparatif des 16 outils de gestion de versions sous licence libre (2)

Catégorie	Critère	CVSNT	Darcs	Git	LibreSource Synchronizer	Mercurial
Information générale	Statut de développement	maintenu et nouvelles fonctionnalités en développement	développement actif	développement actif	maintenu et nouvelles fonctionnalités en développement	développement actif
	Modèle du répertoire	client serveur	distribué	distribué	client serveur	distribué
	Modèle accès simultané	merge or lock	merge	merge	merge	merge
	Licence	GPL ou propriétaire	GPL	GPL	QPL	GPL
	Licence libre	oui ou non	oui	oui	oui	oui
	Portabilité	?	très bon	?	excellent	excellent
	Plateformes supportées	Unix-like, Windows, Mac OS X, OS/400	Unix-like, Windows, Mac OS X	POSIX, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	C++	Haskell	C, shell scripts	Java	Python, C
	Documentation disponible	?	bon	?	moyen	très bon
	Facilité de déploiement	?	très bien	?	excellent	excellent
Fonctionnalités	Validation atomique	oui	oui	oui	oui	oui
	Définir des permissions	?	non	?	permission par répertoire ou branche	oui
	Ensemble de changements	?	oui	?	partiel	oui
	Historique du fichier ligne par ligne	?	oui	?	oui	oui
	Signature des révisions	non	oui	oui	non	oui
	Tags	oui	oui	oui	oui	oui
	Support international	oui	?	?	?	?

Tableau D-7 Comparatif des 16 outils de gestion de versions sous licence libre (3)

Catégorie	Critère	Monotone	OpenCM	Subversion (SVN)	Superversion	SVK	Vesta
Information générale	Statut de développement	développement actif	?	développement actif	?	développement actif	?
	Modèle du répertoire	distribué	?	client serveur	?	distribué	?
	Modèle accès simultané	merge	?	merge or lock	?	merge	?
	Licence	GPL	GPL	Apache-BSD	GPL	Artistic-GPL	GPL
	Licence libre	oui	oui	oui	oui	oui	oui
	Portabilité	excellent	bon: sous UNIX	excellent	excellent	bon	bon
	Plateformes supportées	Unix-like, Windows, Mac OS X		Unix-like, Windows, Mac OS X		Unix-like, Windows, Mac OS X	
Information technique	Langage de programmation	C++	?	C	?	Perl	?
	Documentation disponible	bon	bon: sous UNIX	très bon	pas très bon	pas très bon	très bon
	Facilité de déploiement	excellent	très bon	Il faut Apache 2	très facile	moyen	moyen bien
Fonctionnalités	Validation atomique	oui	oui	oui	oui	oui	oui
	Définir des permissions	oui	par branche	oui	non	oui	oui
	Ensemble de changements	oui	oui	partiel	Partiel	partiel	non
	Historique du fichier ligne par ligne	oui	?	oui	non	oui	non
	Signature des révisions	oui	?	non	?	oui	?
	Tags	oui	?	partiel	?	oui	?
	Support international	?	?	oui	?	oui	?

Tableau D-8 Tableau comparatif de l'ensemble des outils de gestion de versions

Catégorie	Critère	AccuRev	Aegis	Aldon	Alienbrain
Information générale	Statut de développement	développement actif	?	développement actif	développement actif
	Modèle du répertoire	client serveur	?	client serveur	client serveur
	Modèle de l'accès simultané	merge or lock	?	merge or lock	merge or lock
	Licence	propriétaire	GPL	propriétaire	propriétaire
	Licence libre	non	oui	non	non
	Portabilité	excellent	moyen	?	?
	Plateformes supportées	toutes plateformes Java		Linux, Windows, OS/400	Linux, Windows, Mac OS X
Information technique	Langage de programmation	C++, Java	?	C++, Java	?
	Modèle de l'historique	changeset	?	snapshot	changeset
	ID des révisions	namespace	?	namespace	?
	Taille du répertoire	O(revisions)	?	O(revisions)	?
	Documentation disponible	excellent	moyen	?	?
	Facilité de déploiement	excellent	moyen	?	?
	Ensemble de commandes	très étendu	complexe	?	?
	Intégration du réseautage	bon	mauvais	?	?
	Protocoles réseau	custom	?	custom	HTTP, HTTPS
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	oui	oui	oui	oui
	Copie de fichier ou de dossier	oui	non	?	?
	Copie du répertoire	oui	oui	?	?
	Propager les changements	oui	oui	?	?
	Définir des permissions	oui	oui	?	?
	Ensemble de changements	oui	oui	?	?
	Historique du fichier ligne par ligne	oui	oui	?	?

Tableau D-9 Tableau comparatif de l'ensemble des outils de gestion de versions (2)

Catégorie	Critère	AccuRev	Aegis	Aldon	Alienbrain
Fonctionnalités	Possibilité de travailler sur un seul dossier du répertoire	oui	non	?	?
	Possibilité de traquer les changements courants	oui	oui	?	?
	Fusion de fichiers renommés	oui	?	?	?
	Liens symboliques	oui	?	oui	?
	Ancrage pré/post évènement	oui	?	oui	oui
	Signature des révisions	oui	?	oui	?
	Traçabilité des fusions	oui	?	oui	oui
	Caractère de fin de ligne	oui	?	?	?
	Tags	oui	?	?	oui
	Support international	?	?	?	oui
Interfaces utilisateur	Interfaces web	oui	oui	oui	non
	Interfaces graphiques	Windows, Linux, Unix, Mac OS X, BeOS	oui	Windows	windows, Linux, Mac OS X
	Plug-ins pour IDE	IDEA	?	?	Visual studio, Adobe photoshop

Tableau D-10 Tableau comparatif de l'ensemble des outils de gestion de versions (3)

Catégorie	Critère	AllFusion Harvest Change Manager	Arch	Bazaar	BitKeeper
Information générale	Statut de développement	développement actif	maintenu mais pas de nouveau développement	développement actif	Dvp. actif
	Modèle du répertoire	client serveur	distribué	distribué	distribué
	Modèle de l'accès simultané	merge or lock	merge	merge	merge
	Licence	propriétaire	GPL	GPL	propriétaire
	Licence libre	non	oui	oui	non
	Portabilité	?	bon	excellent	très bon
	Plateformes supportées	Unix-like, Windows	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS
Information technique	Langage de programmation	C, Java	C, shell scripts	Python	?
	Modèle de l'historique	changeset et snapshot	changeset	snapshot	?
	ID des révisions	namespace	namespace	pseudorandom	?
	Taille du répertoire	O(patch)	O(patch)	O(patch)	?
	Documentation disponible	?	moyen	excellent	très bon
	Facilité de déploiement	?	excellent	très facile	bon
	Ensemble de commandes	?	beaucoup de commandes compatibles avec CVS	proche CVS sauf lorsque la conception diffère	du type CVS
	Intégration du réseautage	?	Excellent, FTP, SFTP, WebDAV...	excellent	bon
	Protocoles réseau	HTTP, custom (TCP)	WebDAV, HTTP	HTTP, SFTP, FTP, ssh, custom, email bundles	?
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	non	oui	oui	oui
	Copie de fichier ou de dossier	?	non	non	oui
	Copie du répertoire	?	oui	oui	oui
	Propager les changements	?	oui	oui	oui

Tableau D-11 Tableau comparatif de l'ensemble des outils de gestion de versions (4)

Catégorie	Critère	AllFusion Harvest Change Manager	Arch	Bazaar	BitKeeper
Fonctionnalités	Définir des permissions	?	oui	prévu	?
	Ensemble de changements	?	oui	oui	oui
	Historique du fichier ligne par ligne	?	non	oui	oui
	Possibilité de travailler sur un seul dossier du répertoire	?	oui pour commit et non pour vérification	non	non
	Possibilité de traquer les changements courants	?	oui	oui	oui
	Fusion de fichiers renommés	non	?	oui	?
	Liens symboliques	non	oui	oui	?
	Ancrage pré/post évènement	oui	oui	oui	?
	Signature des révisions	non	oui	partiel	?
	Traçabilité des fusions	non	?	oui	?
	Caractère de fin de ligne	oui	?	non	?
	Tags	oui	oui	oui	?
	Support international	oui	?	oui	?
Interfaces utilisateur	Interfaces web	inclus	ArchZoom	Webserve, loggerhead, trac	oui
	Interfaces graphiques	Windows, Linux, Unix	ArchWay, TlaLog	Olive, bzt-gtk (GTK+), QBzr (Qt), TortoiseBZR (Windows)	bon
	Plug-ins pour IDE	Eclipse, Visual studio	?	Eclipse	?

Tableau D-12 Tableau comparatif de l'ensemble des outils de gestion de versions (5)

Catégorie	Critère	ClearCase	CMSynergy	Co-Op	Codeville
Information générale	Statut de développement	développement actif	?	développement actif	développement actif
	Modèle du répertoire	client serveur et distribué	?	distribué	distribué
	Modèle de l'accès simultané	merge or lock	?	merge	merge
	Licence	propriétaire	propriétaire	propriétaire	BSD
	Licence libre	non	non	non	oui
	Portabilité	moyen	très bon	uniquement windows	?
	Plateformes supportées	Unix-like, Windows, i5/OS, z/OS		Windows	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	?	?	C++	?
	Modèle de l'historique	snapshot	?	changeset	?
	ID des révisions	namespace	?	user ID-ordinal	?
	Taille du répertoire	O(patch)	?	O(patch)	?
	Documentation disponible	très bon	moyen	très bon	?
	Facilité de déploiement	mauvais	moyen	très facile	?
	Ensemble de commandes	excellent	étendue et puissante	basique	?
	Intégration du réseautage	mauvais	bon	bon	?
Fonctionnalités	Protocoles réseau	HTTP, custom (CCFS), custom (MVFS filesystem driver)	?	e-mail (MAPI, SMTP/POP3, Gmail), LAN	?
	Validation atomique	oui	oui	oui	?
	Renommage de fichier	oui	oui	oui	?
	Copie de fichier ou de dossier	oui	oui	non	?
	Copie du répertoire	non	oui	oui	?
	Propager les changements	oui	oui	oui	?
	Définir des permissions	oui	non	accès par projet	?
	Ensemble de changements	non	oui	oui	?
Fonctionnalités	Historique du fichier ligne par ligne	oui	probablement	pas directement	?

Tableau D-13 Tableau comparatif de l'ensemble des outils de gestion de versions (6)

Catégorie	Critère	ClearCase	CMSynergy	Co-Op	Codeville
Fonctionnalités	Possibilité de travailler sur un seul dossier du répertoire	oui	oui et non	non	?
	Possibilité de traquer les changements courants	oui	oui	oui	?
	Fusion de fichiers renommés	oui	?	oui	?
	Liens symboliques	oui	?	non	?
	Ancrage pré/post évènement	oui	?	support limité	?
	Signature des révisions	oui	?	non	?
	Traçabilité des fusions	oui	?	non	?
	Caractère de fin de ligne	oui	?	non	?
	Tags	oui	?	oui	?
Interfaces utilisateur	Support international	?	?	?	?
	Interfaces web	inclus	possiblement	pas nécessaire	?
	Interfaces graphiques	Windows, Unix-like	oui	Windows	?
	Plug-ins pour IDE	Eclipse, Visual studio, Kdevelop, IDEA	?	?	?

Tableau D-14 Tableau comparatif de l'ensemble des outils de gestion de versions (7)

Catégorie	Critère	CVS	CVSNT	Darcs	DesignSync
Information générale	Statut de développement	maintenu mais pas de nouveau développement	maintenu et nouvelles fonctionnalités en développement	développement actif	?
	Modèle du répertoire	client serveur	client serveur	distribué	?
	Modèle de l'accès simultané	merge	merge or lock	merge	merge or lock
	Licence	GPL	GPL ou propriétaire	GPL	propriétaire
	Licence libre	oui	oui ou non	oui	non
	Portabilité	bon	?	très bon	?
	Plateformes supportées	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X, OS/400	Unix-like, Windows, Mac OS X	Unix-like, Windows
Information technique	Langage de programmation	C	C++	Haskell	?
	Modèle de l'historique	snapshot	changeset	patch	?
	ID des révisions	namespace	nemaspace	namespace	?
	Taille du répertoire	O(patch)	O(patch)	O(patch)	?
	Documentation disponible	excellent	?	bon	?
	Facilité de déploiement	bon	?	très bien	?
	Ensemble de commandes	simple	?	Commandes compactes et faciles à comprendre	?
	Intégration du réseautage	bon	?	Bon, HTTP...	?
Fonctionnalités	Protocoles réseau	pserver, ssh	sspi, sserver, gserver, pserver, ssh	HTTP, ssh, email	?
	Validation atomique	non	oui	oui	?
	Renommage de fichier	non	oui	oui	?
	Copie de fichier ou de dossier	non	?	non	?
	Copie du répertoire	indirectement	?	oui	?
	Propager les changements	non	?	oui	?
	Définir des permissions	limité	?	non	?
	Ensemble de changements	non	?	oui	?

Tableau D-15 Tableau comparatif de l'ensemble des outils de gestion de versions (8)

Catégorie	Critère	CVS	CVSNT	Darcs	DesignSync
Fonctionnalités	Historique du fichier ligne par ligne	oui	?	oui	?
	Possibilité de travailler sur un seul dossier du répertoire	oui	?	oui pour commit et non pour vérification	?
	Possibilité de traquer les changements courants	oui	?	oui	?
	Fusion de fichiers renommés	non	?	oui	?
	Liens symboliques	non	oui	non	oui
	Ancrage pré/post évènement	support limité	oui	oui	?
	Signature des révisions	non	non	oui	?
	Traçabilité des fusions	non	oui	N/A	?
	Caractère de fin de ligne	oui	oui	?	?
	Tags	oui	oui	oui	oui
	Support international	?	oui	?	?
Interfaces utilisateur	Interfaces web	cvsweb, ViewVC, autres	cvsweb, ViewVC, autres	darcsweb	?
	Interfaces graphiques	TortoiseCVS (Windows explorer), Mac OS X, GTK, Qt	Windows, Mac OS X, OS/400, GTK, Qt	Windows, Mac OS X	?
	Plug-ins pour IDE	Eclipse, Kdevelop, IDEA	comme CVS	?	?

Tableau D-16 Tableau comparatif de l'ensemble des outils de gestion de versions (9)

Catégorie	Critère	Git	LibreSource Synchronizer	Mercurial	Monotone
Information générale	Statut de développement	développement actif	maintenu et nouvelles fonctionnalités en dvp.	développement actif	développement actif
	Modèle du répertoire	distribué	client serveur	distribué	distribué
	Modèle de l'accès simultané	merge	merge	merge	merge
	Licence	GPL	QPL	GPL	GPL
	Licence libre	oui	oui	oui	oui
	Portabilité	?	excellent	excellent	excellent
	Plateformes supportées	POSIX, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	C, shell scripts	Java	Python, C	C++
	Modèle de l'historique	snapshot	changeset	changeset	Hybrid
	ID des révisions	SHA-1 hashes	timestamp of the repository	SHA-1 hashes	SHA-1 hashes
	Taille du répertoire	O(patch)	O(patch)	O(patch)	O(patch)
	Documentation disponible	?	moyen	très bon	bon
	Facilité de déploiement	?	excellent	excellent	excellent
	Ensemble de commandes	?	commandes basiques disponibles, mais utilisation de l'interface préférable	Tente de suivre CVS mais dévie lors de conception différente	Tente de suivre CVS mais dévie lors de conception différente
	Intégration du réseautage	?	bon	Excellent, HTTP ou ssh	Bon, utilise un protocole client appelé netsync
	Protocoles réseau	custom, ssh, rsync, HTTP, FTP, email, bundles	HTTP, file-system	HTTP, ssh, email (with plugin)	custom (netsync)
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	oui	oui	oui	oui

Tableau D-17 Tableau comparatif de l'ensemble des outils de gestion de versions (10)

Catégorie	Critère	Git	LibreSource Synchronizer	Mercurial	Monotone
Fonctionnalités	Copie de fichier ou de dossier	?	non	oui	oui
	Copie du répertoire	?	oui	oui	oui
	Propager les changements	?	oui	oui	oui
	Définir des permissions	?	permission par répertoire ou branche	oui	oui
	Ensemble de changements	?	partiel	oui	oui
	Historique du fichier ligne par ligne	?	oui	oui	oui
	Possibilité de travailler sur un seul dossier du répertoire	?	non	oui pour commit et non pour vérification	oui pour commit et non pour vérification
	Possibilité de traquer les changements courants	?	oui	oui	oui
	Fusion de fichiers renommés	oui	oui	oui	oui
	Liens symboliques	oui	non	oui	non
	Ancrage pré/post événement	oui	support limité	oui	oui
	Signature des révisions	oui	non	oui	oui
	Traçabilité des fusions	oui	oui	oui	oui
	Caractère de fin de ligne	oui	non	oui	non
	Tags	oui	oui	oui	oui
	Support international	?	?	?	?
Interfaces utilisateur	Interfaces web	gitweb, wit, cgit	LibreSource	inclus	ViewMTN, TracMonotone
	Interfaces graphiques	gitk, git-gui, tig, qgit. GTK	Windows, Linux, Unix, Mac OS X	Hgk, TortoiseHg (Windows explorer)	GTK+, Qt
	Plug-ins pour IDE	?	?	Eclipse, NetBeans	?

Tableau D-18 Tableau comparatif de l'ensemble des outils de gestion de versions (11)

Catégorie	Critère	OpenCM	Perforce	PlasticsSCM	PureCM
Information générale	Statut de développement	?	développement actif	développement actif	développement actif
	Modèle du répertoire	?	client serveur	client serveur	client serveur
	Modèle de l'accès simultané	?	merge or lock	merge	merge or lock
	Licence	GPL	propriétaire	propriétaire	propriétaire
	Licence libre	oui	non	non	non
	Portabilité	bon: sous UNIX	excellent	?	bon
	Plateformes supportées		Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	?	?	C#	?
	Modèle de l'historique	?	changeset	changeset	?
	ID des révisions	?	namespace	namespace	?
	Taille du répertoire	?	O(patch)	O(revision)	?
	Documentation disponible	bon: sous UNIX	très bon	?	très bon
	Facilité de déploiement	très bon	très bon	?	très bon
	Ensemble de commandes	du type CVS	très étendue	?	du type CVS
	Intégration du réseautage	bon	bon	?	bon
	Protocoles réseau	?	custom	custom	?
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	oui	oui	oui	oui
	Copie de fichier ou de dossier	non	oui	?	oui
	Copie du répertoire	non	oui	?	non
	Propager les changements	non	?	?	non
	Définir des permissions	par branche	oui	?	oui
	Ensemble de changements	oui	oui	?	oui
	Historique du fichier ligne par ligne	?	oui	?	non
	Possibilité de travailler sur un seul dossier du répertoire	non	oui	?	oui

Tableau D-19 Tableau comparatif de l'ensemble des outils de gestion de versions (12)

Catégorie	Critère	OpenCM	Perforce	PlasticsSCM	PureCM
Fonctionnalités	Possibilité de traquer les changements courants	oui	oui	?	oui
	Fusion de fichiers renommés	?	oui	oui	?
	Liens symboliques	?	oui	oui	?
	Ancrage pré/post évènement	?	oui	oui	?
	Signature des révisions	?	oui	oui	?
	Traçabilité des fusions	?	oui	oui	?
	Caractère de fin de ligne	?	oui	?	?
	Tags	?	oui	oui	?
	Support international	?	oui	?	?
Interfaces utilisateur	Interfaces web	non	inclus	pas nécessaire	non
	Interfaces graphiques	non	Windows, Linux, Unix, Mac OS X, BeOS	Windows, Linux, Unix, Mac OS X	oui
	Plug-ins pour IDE	?	Eclipse, Visual Studio, Kdevelop, IDEA	?	?

Tableau D-20 Tableau comparatif de l'ensemble des outils de gestion de versions (13)

Catégorie	Critère	Razor	SourceAnywhere Hosted	SourceHaven	StarTeam
Information générale	Statut de développement	Dvp. actif	développement actif	Dvp. actif	Dvp. actif
	Modèle du répertoire	client serveur	client serveur	client serveur	client serveur
	Modèle de l'accès simultané	merge or lock	merge or lock	?	merge or lock
	Licence	Concurrent	propriétaire	propriétaire	propriétaire
	Licence libre	non	non	non	non
	Portabilité	?	?	?	?
	Plateformes supportées	Unix-like, Windows, Mac OS X	A SaaS application; Clients: Windows and Cross-platform via Java based client	Unix-like, Windows, Mac OS X	Windows and Cross-platform via Java based client
Information technique	Langage de programmation	C, C++	C++, Java	C, Java	C, Java
	Modèle de l'historique	changeset (texte), snapshot (binary)	changeset	snapshot	snapshot
	ID des révisions	numéros séquentiels	namespace	namespace	MD5 hashes
	Taille du répertoire	Binary (revision), Texte (patch)	O(patch)	O(patch)	O(revision)
	Documentation disponible	?	?	?	?
	Facilité de déploiement	?	?	?	?
	Ensemble de commandes	?	?	?	?
	Intégration du réseautage	?	?	?	?
	Protocoles réseau	TCP/IP	TCP/IP	WebDAV, custom	custom, TCP/IP
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	oui	oui	oui	oui
	Copie de fichier ou de dossier	?	?	?	?
	Copie du répertoire	?	?	?	?
	Propager les changements	?	?	?	?
	Définir des permissions	?	?	?	?
	Ensemble de changements	?	?	?	?

Tableau D-21 Tableau comparatif de l'ensemble des outils de gestion de versions (14)

Catégorie	Critère	Razor	SourceAnywhere Hosted	SourceHaven	StarTeam
Fonctionnalités	Historique du fichier ligne par ligne	?	?	?	?
	Possibilité de travailler sur un seul dossier du répertoire	?	?	?	?
	Possibilité de traquer les changements courants	?	?	?	?
	Fusion de fichiers renommés	oui	non	?	?
	Liens symboliques	non	?	oui	oui
	Ancrage pré/post évènement	oui	oui	oui	non
	Signature des révisions	oui	non	non	non
	Traçabilité des fusions	oui	non	?	oui
	Caractère de fin de ligne	oui	oui	?	oui
	Tags	oui	oui	?	oui
	Support international	non	oui	?	oui
Interfaces utilisateur	Interfaces web	oui, apache, IIS	non	inclus	inclus
	Interfaces graphiques	Unix, Linux, Windows, Mac OS X	Windows, Linux, Unix, Mac OS X, Solaris	Windows, Linux, Unix, Mac OS X, BeOS	Windows, Java, Eclipse, Visual studio, BDS2006 integration
	Plug-ins pour IDE	?	Visual studio, Eclipse, Dreamweaver	?	IDEA

Tableau D-22 Tableau comparatif de l'ensemble des outils de gestion de versions (15)

Catégorie	Critère	Subversion (SVN)	Superversion	Surround SCM	SVK
Information générale	Statut de développement	développement actif	?	développement actif	développement actif
	Modèle du répertoire	client serveur	?	client serveur	distribué
	Modèle de l'accès simultané	merge or lock	?	merge or lock	merge
	Licence	Apache-BSD	GPL	propriétaire	Artistic-GPL
	Licence libre	oui	oui	non	oui
	Portabilité	excellent	excellent	?	bon
	Plateformes supportées	Unix-like, Windows, Mac OS X		Unix-like, Windows, Mac OS X	Unix-like, Windows, Mac OS X
Information technique	Langage de programmation	C	?	C++	Perl
	Modèle de l'historique	changeset et snapshot	?	changeset	changeset
	ID des révisions	namespace	?	namespace	?
	Taille du répertoire	O(patch)	?	O(patch)	?
	Documentation disponible	très bon	pas très bon	?	pas très bon
	Facilité de déploiement	Il faut Apache 2, demande du travail	très facile	?	moyen
	Ensemble de commandes	comme CVS	Besoin de mémoriser l'ensemble de commandes	?	du type CVS
	Intégration du réseautage	Très bon, WebDAV+DeltaV	Bon	?	très bon
	Protocoles réseau	ssh, HTTP et SSL (utilisant WebDAV), custom (svnserve)	?	custom	?
Fonctionnalités	Validation atomique	oui	oui	oui	oui
	Renommage de fichier	oui	non	oui	oui
	Copie de fichier ou de dossier	oui	non	?	oui
	Copie du répertoire	indirectement	oui	?	oui
	Propager les changements	oui	non	?	oui
	Définir des permissions	oui	non	?	oui

Tableau D-23 Tableau comparatif de l'ensemble des outils de gestion de versions (16)

Catégorie	Critère	Subversion (SVN)	Superversion	Surround SCM	SVK
Fonctionnalités	Ensemble de changements	partiel	Partiel	?	partiel
	Historique du fichier ligne par ligne	oui	non	?	oui
	Possibilité de travailler sur un seul dossier du répertoire	oui	non	?	oui
	Possibilité de traquer les changements courants	oui	oui	?	oui
	Fusion de fichiers renommés	non	?	oui	oui
	Liens symboliques	oui	?	non	oui
	Ancrage pré/post évènement	oui	?	oui	oui
	Signature des révisions	non	?	non	oui
	Traçabilité des fusions	non	?	oui	oui
	Caractère de fin de ligne	oui	?	oui	oui
	Tags	partiel	?	oui	oui
	Support international	oui	?	oui	oui
Interfaces utilisateur	Interfaces web	Apache 2 module inclus, WebSVN, ViewSVN, ViewVC, Trac	non	WebDAV	oui
	Interfaces graphiques	Qt, TortoiseSVN (Windows explorer), Java, Mac OS X, Nautilus	oui	Windows, Linux, Unix, Mac OS X	non
	Plug-ins pour IDE	Eclipse, Visual studio, NetBeans, IDEA, Kdevelop	?	IDEA, Eclipse, Visual studio	?

Tableau D-24 Tableau comparatif de l'ensemble des outils de gestion de versions (17)

Catégorie	Critère	Team Foundation Server	Telelogic Synergy	Vault	Vesta	Visual SourceSafe
Information générale	Statut de développement	développement actif	développement actif	développement actif	?	uniquement correction des gros bugs
	Modèle du répertoire	client serveur	client serv. et dist.	client serveur	?	client serveur
	Modèle de l'accès simultané	merge or lock	merge or lock	merge or lock	?	merge or lock
	Licence	propriétaire	propriétaire	propriétaire	GPL	propriétaire
	Licence libre	non	non	non	oui	non
	Portabilité	?	?	?	bon	uniquement windows
	Plateformes supportées	Server: Windows Server 2003; Clients Windows	Linux, Windows, Unix-like	Unix-like, Linux, Windows		Windows
Information technique	Langage de programmation	C++, C#	Java	C#	?	C
	Modèle de l'historique	changeset	changeset (texte), snapshot (binary)	snapshot	?	snapshot
	ID des révisions	namespace	namespace	?	?	namespace ?
	Taille du répertoire	O(patch)	O(patch)	O(patch)	?	O(changes)?
	Documentation disponible	?	?	?	très bon	moyen
	Facilité de déploiement	?	?	?	moyen bien	très bien
	Ensemble de commandes	?	?	?	moyen bien	moyen
	Intégration du réseautage	?	?	?	très bon	bon
Fonctionnalités	Protocoles réseau	SOAP sur HTTP ou HTTPS	HTTP, ssh, custom	HTTP, HTTPS	?	aucun
	Validation atomique	oui	oui	oui	oui	non
	Renommage de fichier	oui	oui	oui	oui	oui
	Copie de fichier ou de dossier	?	?	?	oui	oui
	Copie du répertoire	?	?	?	oui	non
	Propager les changements	?	?	?	oui	non

Tableau D-25 Tableau comparatif de l'ensemble des outils de gestion de versions (18)

Catégorie	Critère	Team Foundation Server	Telelogic Synergy	Vault	Vesta	Visual SourceSafe
Fonctionnalités	Définir des permissions	?	?	?	oui	oui
	Ensemble de changements	?	?	?	non	non
	Historique du fichier ligne par ligne	?	?	?	non	pas directement
	Possibilité de travailler sur un seul dossier du répertoire	?	?	?	oui	oui
	Possibilité de traquer les changements courants	?	?	?	oui	oui
	Fusion de fichiers renommés	oui	oui	oui	?	?
	Liens symboliques	?	oui	non	?	oui
	Ancrage pré/post événement	oui	oui	oui	?	oui
	Signature des révisions	?	oui	non	?	non
	Traçabilité des fusions	oui	oui	non	?	non
	Caractère de fin de ligne	?	oui	oui	?	?
	Tags	oui	oui	oui	?	oui
	Support international	oui	oui	?	?	?
Interfaces utilisateur	Interfaces web	inclus	via Telelogic Change interface	inclus	oui	pas inclus, SSWI, VSS Remoting
	Interfaces graphiques	Windows, Mac OS, Unix	Windows, Linux, Unix, Java	Windows, Unix-like, MaC OS X	non	Windows, Linux, Mac OS, Solaris, Machine virtuelle Java utilisant SourceAnywhere
	Plug-ins pour IDE	Visual studio, client Java pour Eclipse IDE	Eclipse, Visual studio, IDEA	?	?	Visual Studio, IDEA

APPENDICE. E - ANALYSE DE LIBRE SOURCE

LibreSourceSynchroniser est un outil appartenant à la plateforme LibreSource qui est une plateforme collaborative polyvalente. Cette plateforme sous licence libre est présentée comme étant modulaire, hautement personnalisable et adaptée au développement de logiciel collaboratif, au collecticiel⁴, à l'animation de communauté, à l'archivage électronique et à l'édition pour le Web. LibreSource permet d'accueillir différents projets et utilisateurs sur un même serveur. Il s'agit d'un site web modulable que ses utilisateurs peuvent personnaliser en ligne en combinant des ressources avec des droits d'accès. Les ressources proposées dans LibreSource sont : wiki, forum, traceur de *bug* (*bug tracker*), synchroniseur, fichiers, espace de téléchargement, espace de téléversement, formulaire, messagerie instantanée, notification de l'utilisateur par courriel, sécurité, importation et exportation en XML, questionnaire, etc. De plus, LibreSource permet de gérer des utilisateurs et de leur octroyer des droits différents.

Cette plateforme nous a semblé pouvoir être pertinente pour notre projet. Son utilisation permettrait plus que la gestion de versions, la plateforme pourrait servir de base pour notre système. En effet, après une première exploration rapide, nous avons pu noter cinq éléments que cet outil comporte et qui seraient utiles pour notre projet. D'abord, LibreSource présente une interface web pour ses utilisateurs. De plus, la gestion des utilisateurs et des groupes d'utilisateurs y est intégrée. LibreSource permet aussi de créer un répertoire de fichiers (pour les productions des apprenants dans le cas de notre projet) et contient la notion d'évènements (ce qui semble être une bonne piste pour la gestion des multi-utilisateurs). Pour finir, cette plateforme

⁴ Logiciel permettant à des utilisateurs reliés par un réseau de travailler en collaboration sur un même projet (Grand dictionnaire terminologique de l'Office québécois de la langue française)

contient un système de gestion de versions. Ces cinq éléments nous ont séduits et nous avons donc installé un serveur LibreSource afin de pouvoir tester plus en détail ses fonctionnalités et étudier s'il serait possible d'utiliser cette plateforme dans notre projet.

Si nous avons choisi d'utiliser LibreSource, alors chaque modèle de l'apprenant devrait être vu comme un projet (de LibreSource). Chaque projet peut être constitué de différentes ressources contenues dans la plateforme et il est possible de spécifier des droits d'accès à des groupes d'utilisateurs. Ainsi pour notre modèle de l'apprenant, le projet serait constitué d'un répertoire de compétences, d'un répertoire de fichiers (les productions de l'apprenant) et de liens entre les éléments de ces deux répertoires. Pour la gestion de versions, nous utiliserions l'outil LibreSourceSynchroniser qui est intégré à la plateforme.

Plusieurs des ressources proposées et intégrées dans LibreSource sont utiles à notre projet de modélisation de l'apprenant. Nous avons déjà cité les cinq principales qui ont attiré notre attention sur cette plateforme soit : (1) une interface web, (2) la gestion des utilisateurs et de groupes d'utilisateurs, (3) répertoire de fichiers (pour les productions des apprenants), (4) notion d'évènements (pourrait permettre d'envoyer des notifications aux acteurs concernés lorsqu'une modification est apportée à un modèle) et (5) LibreSourceSynchroniser, un système de gestion de versions. D'autres ressources intégrées à la plateforme LibreSource peuvent nous être utiles. Comme par exemple l'exportation en XML qui permet d'exporter les modèles d'apprenant en version XML pour des fins d'interopérabilité. Il existe aussi la possibilité de téléverser et de télécharger des fichiers dans un répertoire, ainsi les productions des apprenants peuvent être déposées par ceux-ci mais aussi récupérées par d'autres acteurs.

Pour tester cette plateforme, nous avons simulé notre projet sur le serveur que nous avons installé. Nous avons donc créé des projets (chacun étant un modèle d'apprenant), des utilisateurs et des groupes d'utilisateurs. Nous avons ensuite essayé de répondre aux besoins de notre projet avec les ressources proposées dans

LibreSource. Différentes difficultés sont alors apparues. Premièrement, l'interface n'est pas facile à utiliser pour un non informaticien. Effectivement, LibreSource semble avoir été développé en premier lieu pour la gestion de projets collaboratifs en informatique et plus précisément en développement de code informatique. Comme nous l'avons déjà dit, de nombreux outils de gestion de versions sont développés dans le cadre de projet de développement de code informatique ce qui semble être le cas pour LibreSource. Bien sûr, il est vrai, pour LibreSource comme pour les outils de gestion de versions, qu'il est possible d'utiliser ces systèmes pour d'autres types de fichiers. Toutefois, dans le cas de projet de développement de code informatique, les utilisateurs visés de la plateforme appartiennent tous à un même groupe d'utilisateurs, des informaticiens participant à un projet commun. Dans notre cas, les utilisateurs potentiels de notre système sont très diversifiés et surtout n'ont pas tous des connaissances en développement informatique. Une interface peut paraître très simple à un développeur mais très compliquée pour un novice. Dans le cas de LibreSource, la structure de la plateforme et de la navigation web est la même que celle des projets, c'est-à-dire que c'est une structure en arbre. Notre modèle de l'apprenant n'est pas structuré en arbre, mais plutôt composé de différents groupes d'éléments (des compétences d'un côté, des productions de l'autre, des liens entre les deux...). En utilisant LibreSource nous aurions été obligés de créer une structure en arbre et l'interface de notre système devrait suivre cette structure en arbre qui n'est absolument pas intuitive pour un utilisateur final du modèle de l'apprenant. Un des éléments qui nous avait séduits pour le choix de LibreSource, l'existence d'une interface web, perd sa valeur puisque cette interface devrait être repensée selon les besoins de notre projet et afin de répondre aux caractéristiques de nos utilisateurs finaux (principalement le fait qu'ils ne seront pas obligatoirement des informaticiens).

De plus, un autre problème décelé dans LibreSource est lié à la nature même de notre modèle de l'apprenant. En effet, celui-ci emprunte une approche par compétences. Comme il n'existe pas de ressource compétence dans la plateforme étudiée, il faudrait en ajouter une sur laquelle on pourrait fonder le développement de

nos modèles d'apprenants. Après investigation, nous avons découvert que l'ajout d'une ressource dans la plateforme nécessite des connaissances informatiques et techniques sur :

- la technologie Enterprise JavaBeans (EJB) qui est une architecture de composants logiciels côté serveur;
- sur Java Open Application Server (JOnAS) qui est un serveur d'application sous licence libre multi-systèmes d'exploitation, et multi bases de données;
- sur Apache Ant qui est un projet sous licence libre visant le développement d'un logiciel d'automatisation des opérations répétitives tout au long du cycle de développement logiciel;
- sur le JavaServer Pages (JSP) qui est une technologie permettant de générer dynamiquement du code HTML, XML ou tout autre type de page web;
- et finalement sur le langage de programmation Java.

Il nous est apparu que le temps nécessaire pour s'appropriier ces différents outils informatiques serait trop long. Le gain de temps que nous pourrions obtenir par l'utilisation de LibreSource grâce à la réutilisation de matériel existant serait en grande partie perdu par le temps nécessaire à acquérir les connaissances sur les technologies nécessaires à la modification de la plateforme pour les besoins de notre projet. Nous avons donc décidé de ne pas utiliser LibreSource dans le cadre de notre projet.

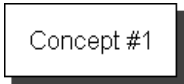
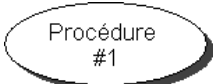
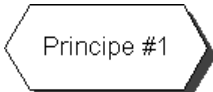
APPENDICE. F - MODÉLISATION PAR OBJETS TYPÉS (LÉGENDE DU LOGICIEL MOT)

La technique MOT a été élaborée par (Paquette, 2002a). Elle utilise, entre autres, trois types d'objet abstraits (concept, procédure, principe) ainsi que des liens entre ces objets (inquant/produit, spécialisation, composition, composition multiple, régulation, précedence). Nous présentons ces objets et liens dans cet appendice.

F.1 Type d'objet MOT standard

Dans le modèle de type MOT standard, les connaissances abstraites sont classées en trois catégories principales (Concept, Procédure et Principe). Chacun de ces objets est représenté à l'écran par une forme graphique différente. Le tableau suivant illustre l'interprétation des types d'objet et de la forme graphique lui étant associée.

Tableau F-1. Interprétation des types d'objet en fonction des formes graphiques

Type	Interprétation
Concept 	Décrit la nature des entités d'un domaine – le quoi . - classes d'objets ou d'événements, types de documents, catégories d'outils, catégories des personnes, ...
Procédure 	Décrit des opérations permettant d'agir sur les concepts – le comment . - opérations génériques, tâches et activités, instructions, scénarios, ...
Principe 	Décrit les contraintes, les conditions ou les acteurs d'une procédure (le pourquoi, le quand et le qui) - propriétés, contraintes, relations de cause à effet, lois, règles de décision, prescription, agent régulateur, acteur, ... <u>Avec le signe A</u> : un acteur qui régit une procédure. <u>Avec le signe O</u> : un outil qui supporte une procédure.

F.2 Type des liens entre objet MOT standard

Les divers types d'objets peuvent être liés entre eux par différents types de liens. Nous donnons un aperçu dans le tableau suivant.

Tableau F-2. Interprétation des types de liens

Type de lien	Interprétation
I/P (intransit-produit)	Met en relation : - un concept entrant à une procédure (le concept est une ressource à la procédure) - une procédure à un concept sortant (le concept est un produit de la procédure)
S (spécialisation)	Met en relation deux objets standard de même type dont l'un est « une sorte de », un cas particulier de l'autre.
C (composition)	Met en relation un type d'objet à l'une de ses composantes ou de ses parties constitutives.
C* (composition multiple)	Met en relation un type d'objet à plusieurs composantes de même type.
R (régulation)	Met en relation un principe qui conditionne un concept, une procédure ou un autre principe. Relie aussi un acteur à la procédure qu'il doit accomplir, ou un outil à la procédure qu'il doit supporter.
P (précédence)	Met en relation deux procédures ou principes dont le premier doit être terminé ou évalué avant que le deuxième ne commence.

APPENDICE. G - FICHIER DE *MAPPING* DE CASTOR

Le fichier de *mapping* que nous avons créé pour Castor est un fichier XML. Il permet à Castor de « traduire » les objets Java en XML et inversement.

```
<?xml version="1.0"?>
<!DOCTYPE mapping PUBLIC "-//EXOLAB/Castor Mapping DTD Version
1.0//EN" "http://castor.org/mapping.dtd">
<mapping>
  <class name="composant.modele.Modele">
    <map-to xml="Modele" />
    <field name="ID" type="string">
      <bind-xml name="ID" node="attribute" />
    </field>
    <field name="acteur" type="composant.acteur.Acteur">
      <bind-xml name="Acteur" />
    </field>
    <field name="listeCompetence"
type="composant.competence.Competence" collection="arraylist">
      <bind-xml name="Competence" />
    </field>
    <field name="listeExpProf" type="composant.acteur.ExpProf"
collection="arraylist">
      <bind-xml name="ExpProf" />
    </field>
    <field name="listeProd" type="composant.acteur.Production"
collection="arraylist">
      <bind-xml name="Production" />
    </field>
    <field name="listeLienExpProf" type="composant.modele.LienExpProf"
collection="arraylist">
      <bind-xml name="LienExpProf" />
    </field>
    <field name="listeLienProd" type="composant.modele.LienProd"
collection="arraylist">
      <bind-xml name="LienProd" />
    </field>
```

```

        <field name="listePointDeVue" type="composant.modele.PointDeVue"
collection="arraylist">
            <bind-xml name="PointDeVue" />
        </field>
    </class>
    <class name="composant.acteur.Acteur">
        <field name="ID" type="string">
            <bind-xml name="ID" node="attribute" />
        </field>
        <field name="statut" type="string">
            <bind-xml name="statut" node="attribute" />
        </field>
        <field name="nom" type="string">
            <bind-xml name="Nom" node="element" />
        </field>
        <field name="prenom" type="string">
            <bind-xml name="Prenom" node="element" />
        </field>
        <field name="adresse" type="string">
            <bind-xml name="Adresse" node="element" />
        </field>
        <field name="telephone" type="string">
            <bind-xml name="Telephone" node="element" />
        </field>
        <field name="mail" type="string">
            <bind-xml name="Mail" node="element" />
        </field>
        <field name="occupation" type="string">
            <bind-xml name="occupation" node="element" />
        </field>
    </class>
    <class name="composant.competence.Competence">
        <field name="ID" type="string">
            <bind-xml name="ID" node="attribute" />
        </field>
        <field name="nom" type="string">
            <bind-xml name="Nom" node="element" />
        </field>
        <field name="description" type="string">
            <bind-xml name="Description" node="element" />
        </field>
        <field name="auteur" type="composant.acteur.Acteur">
            <bind-xml name="Auteur" />
        </field>
    </class>

```

```

</field>
<field name="dateAjout" type="string">
  <bind-xml name="DateAjout" node="element" />
</field>
<field name="habilete" type="composant.competence.Habilete">
  <bind-xml name="Habilete" />
</field>
<field name="connaissance" type="string">
  <bind-xml name="Connaissance" node="element" />
</field>
<field name="ontologie" type="string">
  <bind-xml name="Ontologie" node="element" />
</field>
<field name="niveau" type="composant.competence.Niveau">
  <bind-xml name="Niveau" />
</field>
<field name="categorieContexte" type="int">
  <bind-xml name="categorieContexte" node="element" />
</field>
<field name="nomContexte" type="string">
  <bind-xml name="nomContexte" node="element" />
</field>
</class>
<class name="composant.competence.Habilete">
  <field name="ID" type="string">
    <bind-xml name="ID" node="attribute" />
  </field>
  <field name="nom" type="string">
    <bind-xml name="Nom" node="element" />
  </field>
  <field name="nomReferentiel" type="string">
    <bind-xml name="NomReferentiel" node="element" />
  </field>
  <field name="rang" type="int">
    <bind-xml name="Rang" node="element" />
  </field>
</class>
<class name="composant.competence.Niveau">
  <field name="ID" type="string">
    <bind-xml name="ID" node="attribute" />
  </field>
  <field name="nom" type="string">
    <bind-xml name="Nom" node="element" />
  </field>

```



```

    </field>
    <field name="referentielNiveau" type="string">
        <bind-xml name="ReferentielNiveau" node="element" />
    </field>
    <field name="rang" type="double">
        <bind-xml name="Rang" node="element" />
    </field>
</class>
<class name="composant.acteur.ExpProf">
    <field name="ID" type="string">
        <bind-xml name="ID" node="attribute" />
    </field>
    <field name="nom" type="string">
        <bind-xml name="Nom" node="element" />
    </field>
    <field name="dateDebut" type="string">
        <bind-xml name="DateDebut" node="element" />
    </field>
    <field name="dateFin" type="string">
        <bind-xml name="DateFin" node="element" />
    </field>
    <field name="dateAjout" type="string">
        <bind-xml name="DateAjout" node="element" />
    </field>
    <field name="description" type="string">
        <bind-xml name="Description" node="element" />
    </field>
    <field name="nomEntreprise" type="string">
        <bind-xml name="NomEntreprise" node="element" />
    </field>
</class>
<class name="composant.acteur.Production">
    <field name="ID" type="string">
        <bind-xml name="ID" node="attribute" />
    </field>
    <field name="nom" type="string">
        <bind-xml name="Nom" node="element" />
    </field>
    <field name="description" type="string">
        <bind-xml name="Description" node="element" />
    </field>
    <field name="adresse" type="string">
        <bind-xml name="Adresse" node="element" />
    </field>

```

```

</field>
<field name="dateCreation" type="string">
  <bind-xml name="DateCreation" node="element" />
</field>
<field name="dateAjout" type="string">
  <bind-xml name="DateAjout" node="element" />
</field>
<field name="categorieContexte" type="int">
  <bind-xml name="categorieContexte" node="element" />
</field>
<field name="nomContexte" type="string">
  <bind-xml name="nomContexte" node="element" />
</field>
<field name="tamponee" type="composant.acteur.Acteur">
  <bind-xml name="tamponee" node="element" />
</field>
</class>
<class name="composant.modele.LienExpProf">
  <field name="ID" type="string">
    <bind-xml name="ID" node="attribute" />
  </field>
  <field name="IDCompetence" type="string">
    <bind-xml name="IDCompetence" node="element" />
  </field>
  <field name="IDExpProf" type="string">
    <bind-xml name="IDExpProf" node="element" />
  </field>
  <field name="categorieContexte" type="int">
    <bind-xml name="categorieContexte" node="element" />
  </field>
  <field name="nomContexte" type="string">
    <bind-xml name="nomContexte" node="element" />
  </field>
  <field name="auteur" type="composant.acteur.Acteur">
    <bind-xml name="auteur" node="element" />
  </field>
  <field name="tamponee" type="composant.acteur.Acteur">
    <bind-xml name="tamponee" node="element" />
  </field>
</class>
<class name="composant.modele.LienProd">
  <field name="ID" type="string">
    <bind-xml name="ID" node="attribute" />

```

```

</field>
<field name="IDCompetence" type="string">
  <bind-xml name="IDCompetence" node="element" />
</field>
<field name="IDProduction" type="string">
  <bind-xml name="IDProd" node="element" />
</field>
<field name="categorieContexte" type="int">
  <bind-xml name="categorieContexte" node="element" />
</field>
<field name="nomContexte" type="string">
  <bind-xml name="nomContexte" node="element" />
</field>
<field name="auteur" type="composant.acteur.Acteur">
  <bind-xml name="auteur" node="element" />
</field>
<field name="tamponee" type="composant.acteur.Acteur">
  <bind-xml name="tamponee" node="element" />
</field>
</class>
<class name="composant.modele.PointDeVue">
  <field name="ID" type="string">
    <bind-xml name="ID" node="attribute" />
  </field>
  <field name="nom" type="string">
    <bind-xml name="Nom" node="element" />
  </field>
  <field name="listeCompetence"
type="composant.competence.Competence" collection="arraylist">
    <bind-xml name="Competence" />
  </field>
  <field name="listeLienExpProf" type="composant.modele.LienExpProf"
collection="arraylist">
    <bind-xml name="LienExpProf" />
  </field>
  <field name="listeLienProd" type="composant.modele.LienProd"
collection="arraylist">
    <bind-xml name="LienProd" />
  </field>
  <field name="possesseurPDV" type="composant.acteur.Acteur">
    <bind-xml name="possesseurPDV"/>
  </field>
  <field name="IDStatut" type="string">

```

```
        <bind-xml name="IDStatut" node="element" />
    </field>
</class>
</mapping>
```

APPENDICE. H - FICHIER XSL

Le fichier XSL est un fichier XML, une feuille de style, qui permet d'afficher des XML.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8" indent="yes" />
<xsl:template match="/">
    <HTML>
        <HEAD>
            <TITLE>Modèle de l'apprenant</TITLE>
            <!-- Ajouter ici le lien vers la feuille de style CSS -->
        </HEAD>
        <BODY>
            <xsl:apply-templates/>
        </BODY>
    </HTML>
</xsl:template >
<xsl:template match="Acteur" >
    <div class="Acteur">
        <FONT size="6"><b>Modèle de <xsl:value-of select="Prenom"/>
<xsl:text> </xsl:text> <xsl:value-of select="Nom"/></b></FONT>
        <br />
    </div>
</xsl:template >
<xsl:template match="ExpProf" >
    <div class="ExpProf">
        <br />
        <FONT size="4"><b>Expérience professionnelle : </b> <xsl:value-of
select="Nom"/></FONT>
        <br />
        <b>Description : </b> <xsl:value-of select="Description"/>
        <br />
        <b>Nom de l'entreprise : </b> <xsl:value-of select="NomEntreprise"/>
        <br />
    </div>
</xsl:template >
```

```

        <b>Date début : </b> <xsl:value-of select="DateDebut"/>
        <br />
        <b>Date fin : </b> <xsl:value-of select="DateFin"/>
        <br />
        <b>Date Ajout : </b> <xsl:value-of select="DateAjout"/>
        <br />
    </div>
</xsl:template >
<xsl:template match="Production" >
    <div class="Production">
        <br />
        <FONT size="4"><b>Production : </b> <xsl:value-of
select="Nom"/></FONT>
        <br />
        <b>Description : </b><xsl:value-of select="Description"/>
        <br />
        <b>Adresse : </b><xsl:value-of select="Adresse"/>
        <br />
        <b>Date création : </b><xsl:value-of select="DateCreation"/>
        <br />
        <b>Date ajout : </b><xsl:value-of select="DateAjout"/>
        <br />
        <b>Contexte </b> <xsl:value-of select="nomContexte"/> (catégorie
<xsl:value-of select="categorieContexte"/>)
        <br />
        <xsl:for-each select="tamponee">
            <div id="@id">
                <b>Tamponée/validée par </b> <xsl:value-of
select="Prenom" /> <xsl:text> </xsl:text> <xsl:value-of select="Nom" />
            </div>
        </xsl:for-each>
    </div>
</xsl:template >
<xsl:template match="LienExpProf" >
    <div class="LienExpProf">
        <br /><b>Lien entre la compétence </b><xsl:value-of
select="IDCompetence"/> <b> et l'expérience professionnelle </b><xsl:value-of
select="IDExpProf"/>
        <br />
        <b>Contexte </b> <xsl:value-of select="nomContexte"/> (catégorie
<xsl:value-of select="categorieContexte"/>)
        <xsl:for-each select="auteur">
            <div id="@id">

```

```

        <b>Ajouté par </b> <xsl:value-of select="Prenom" />
<xsl:text> </xsl:text> <xsl:value-of select="Nom" />
        </div>
    </xsl:for-each>
    <xsl:for-each select="tamponee">
        <div id="@id">
            <b>Tamponé/validé par </b> <xsl:value-of select="Prenom"
/> <xsl:text> </xsl:text> <xsl:value-of select="Nom" />
            </div>
        </xsl:for-each>
    </div>
</xsl:template >
<xsl:template match="LienProd" >
    <div class="LienProd">
        <br /><b>Lien entre la compétence </b><xsl:value-of
select="IDCompetence"/> <b> et la production </b><xsl:value-of select="IDProd"/>
        <br />
        <b>Contexte </b> <xsl:value-of select="nomContexte"/> (catégorie
<xsl:value-of select="categorieContexte"/>)
        <br />
        <xsl:for-each select="auteur">
            <div id="@id">
                <b>Ajouté par </b> <xsl:value-of select="Prenom" />
<xsl:text> </xsl:text> <xsl:value-of select="Nom" />
                </div>
            </xsl:for-each>
            <xsl:for-each select="tamponee">
                <div id="@id">
                    <b>Tamponé/validé par </b> <xsl:value-of select="Prenom"
/> <xsl:text> </xsl:text> <xsl:value-of select="Nom" />
                    </div>
                </xsl:for-each>
            </div>
        </xsl:template >
<xsl:template match="Competence" >
    <div class="Competence">
        <br />
        <FONT size="4"><b>Compétence <xsl:value-of
select="Nom"/></b></FONT>
        <br />
        <b>Description : </b><xsl:value-of select="Description"/>
        <br />
        <xsl:for-each select="Auteur">

```

```

        <div id="@id">
            <b>Auteur : </b><xsl:value-of select="Prenom" /> <xsl:text>
</xsl:text> <xsl:value-of select="Nom" />
        </div>
    </xsl:for-each>

    <xsl:for-each select="Habilete">
        <div id="@id">
            <b>Habilité : </b> <i><xsl:value-of select="Nom" /></i> du
référentiel <i><xsl:value-of select="NomReferentiel" /></i> , rang <i><xsl:value-of
select="Rang" /></i>
        </div>
    </xsl:for-each>
    <b>Connaissance : </b><xsl:value-of select="Connaissance"/>
    <br />
    <b>Ontologie : </b><xsl:value-of select="Ontologie"/>
    <br />
    <xsl:for-each select="Niveau">
        <div id="@id">
            <b>Niveau : </b><xsl:value-of select="Nom" /> du
référentiel <i><xsl:value-of select="ReferentielNiveau" /></i> , rang <i><xsl:value-
of select="Rang" /></i>
        <br />
    </div>
    </xsl:for-each>
    <b>Contexte </b> <xsl:value-of select="nomContexte"/> (catégorie
<xsl:value-of select="categorieContexte"/>)
    <br />
</div>
</xsl:template >
<xsl:template match="PointDeVue" >
    <div class="PointDeVue">
        <br />
        <FONT size="5"><b><xsl:value-of select="Nom"/></b></FONT>
        <br />
        <xsl:for-each select="Competence" >
            <div id="@id">
                <br />
                <FONT size="4"><b>Compétence <xsl:value-of
select="Nom"/></b></FONT>
                <br />
                <b>Description : </b><xsl:value-of select="Description"/>
                <br />
            </div>
        </xsl:for-each>
    </div>
</xsl:template>

```



```

        <xsl:for-each select="Auteur">
        <div id="@id">
            <b>Auteur : </b><xsl:value-of select="Prenom" /> <xsl:text>
</xsl:text> <xsl:value-of select="Nom" />
            </div>
        </xsl:for-each>
        <xsl:for-each select="Habilete">
            <div id="@id">
                <b>Habilité : </b> <i><xsl:value-of select="Nom" /></i> du
référentiel <i><xsl:value-of select="NomReferentiel" /></i> , rang <i><xsl:value-of
select="Rang" /></i>
            </div>
        </xsl:for-each>
        <b>Connaissance : </b><xsl:value-of select="Connaissance"/>
        <br />
        <b>Ontologie : </b><xsl:value-of select="Ontologie"/>
        <br />
        <xsl:for-each select="Niveau">
            <div id="@id">
                <b>Niveau : </b><xsl:value-of select="Nom" /> du
référentiel <i><xsl:value-of select="ReferentielNiveau" /></i> , rang <i><xsl:value-
of select="Rang" /></i>
            <br />
            </div>
        </xsl:for-each>
        <b>Contexte </b> <xsl:value-of select="nomContexte"/> (catégorie
<xsl:value-of select="categorieContexte"/>)
        <br />
    </div>
</xsl:for-each >
</div>
</xsl:template >
</xsl:stylesheet>

```

APPENDICE. I - DÉTAILS DES SIMULATIONS

Cette annexe contient la description intégrale des trois simulations et de leurs résultats.

I.1. Simulation 1 : cours en ligne avec acteurs humains

Dans cette simulation, nous nous situons dans un contexte similaire au contexte d'un cours universitaire, mais pour les besoins de la simulation nous modifierons le contexte original dont nous nous inspirons (présenté au paragraphe qui suit). Comme nous allons l'expliquer, dans ce contexte, les professeurs et tuteurs sont les évaluateurs des compétences et les décideurs, c'est-à-dire que la stratégie de négociation utilisée se fonde sur leurs évaluations. C'est la première stratégie décrite au chapitre IV page 112.

Dans ce scénario, on veut faire la preuve d'une part du bon fonctionnement de notre système et d'autre part de la fonction négociation. Pour la négociation, nous montrerons celle-ci dans tous les cas de figure pouvant se présenter (voir chapitre IV). Concernant notre système, nous montrerons ici :

- l'utilisation de points de vue
- l'ajout d'une compétence, d'une production
- la preuve d'une compétence par une production
- la certification d'un tel lien
- la fermeture du modèle contexte temporaire (soit la négociation de toutes les compétences et la conservation du point de vue de l'apprenant).

I.1.1 Contexte

I.1.1.1. Description du contexte

Nous nous inspirons du cours universitaire INF 4018 intitulé *Projet d'intégration* fondé sur une activité d'autoévaluation des compétences du professeur Richard Hotte à la Télé-université. Ce cours est un cours-projet permettant aux étudiants de faire la synthèse des connaissances et des habiletés développées au cours du programme (ce cours a été développé pour deux programmes : le certificat en informatique appliquée à l'organisation et le certificat en informatique appliquée). Plus précisément, le cours permet aux étudiants de faire le bilan de leurs compétences techniques, professionnelles et personnelles en vue de les utiliser dans la réalisation d'un projet professionnel (Hotte, 2009). Pour effectuer ce bilan, un référentiel de compétences du programme au complet a été développé. C'est pourquoi nous avons choisi ce cours comme contexte de notre première simulation, avec l'autorisation du professeur Hotte.

I.1.1.2. Référentiel de compétences

Le référentiel de compétences du cours du professeur Hotte a été développé avec la définition de compétence de Paquette (2002a), avec son référentiel d'habiletés génériques et avec ses indicateurs de performance (présentés au chapitre I).

Nous n'allons pas utiliser l'ensemble du référentiel de compétences du cours du professeur Hotte car celui-ci est très complet et nous n'avons pas besoin d'autant de compétences pour effectuer notre simulation. Pour notre simulation, nous avons choisi un ensemble de 11 compétences organisées en quatre catégories : compétence technique – générale, compétence professionnelle – communiquer, compétence professionnelle – travailler avec les autres et compétence personnelle. Le tableau I-1 présente les compétences que nous avons choisies et qui forment le référentiel que nous allons utiliser dans cette première simulation.

Dans notre modèle de l'apprenant, une compétence n'est pas liée à une catégorie mais plutôt à une ontologie. Dans le cadre de cette simulation, nous n'utilisons pas cet aspect du modèle (il sera démontré par une autre simulation).

Tableau I-1. Référentiel de compétences du contexte 1 (Professeur Hotte)

Catégorie	Nom	Énoncé de la compétence	Connaissance	Habilité
Compétence technique - Générale	C1	Identifier ses connaissances préalables du domaine étudié	Techniques d'auto-évaluation	Identifier (2)
	C2	Planifier ses apprentissages	Techniques d'auto-gestion	Synthétiser (8)
	C3	Appliquer les techniques pour configurer l'environnement logiciel	Techniques de configuration et composants de l'environnement	Appliquer (5)
	C4	Appliquer les techniques de communication avec le tuteur et les pairs à l'aide des outils proposés dans le cours	Techniques de communication et outils de communication	Appliquer (5)
	C5	Auto-évaluer ses apprentissages	Techniques de mesure	Auto-contrôler (10)
Compétence prof. – Communiquer	C6	Exprimer clairement ses idées à d'autres personnes et les appuyer à l'aide d'éléments pratiques et théoriques	Techniques de communication	Synthétiser (8)
	C7	Synthétiser des informations de natures variées, à l'aide d'un support écrit, ou à l'oral, de façon à atteindre son objectif d'information ou de communication vers des interlocuteurs concernés	Techniques de communication	Synthétiser (8)
Compétence prof. - Travailler avec les autres	C8	Synthétiser ses idées en groupe en prêtant attention aux idées des autres	Techniques d'animation de groupe	Synthétiser (8)
	C9	S'adapter (autocontrôler) aux pensées et aux opinions des autres dans un groupe	Techniques d'animation de groupe	Auto-contrôler (10)
Compétence personnelle	C10	Prêter attention aux autres	Techniques d'animation de groupe	Prêter attention (1)
	C11	Influencer ses interlocuteurs internes et externes afin de parvenir à un accord sur un sujet donné.	Stratégies de communication	Auto-contrôler (10)

I.1.1.3. Référentiels d'habiletés et de niveaux

Les référentiels utilisés dans ce contexte sont ceux de Paquette (2002a). Rappelons ceux-ci brièvement. Le référentiel d'habiletés comporte 10 habiletés génériques ordonnées par ordre ascendant (1. Porter attention, 2. Intégrer, 3. Instancier/préciser, 4. Transposer/traduire, 5. Appliquer, 6. Analyser, 7. Réparer, 8. Synthétiser, 9. Évaluer, 10. Auto-contrôler).

Le référentiel de niveaux utilisé correspond aux indicateurs de performance de Paquette (2002a). Le référentiel de Paquette comporte cinq indicateurs et quatre niveaux (voir tableau I-2 du chapitre I page 34). Ces quatre niveaux sont du plus bas au plus élevé sensibilisation (niveau 1), familiarisation (niveau 2), maîtrise (niveau 3) et expertise (niveau 4).

I.1.1.4. Contrat

Comme nous l'avons spécifié au chapitre III (page 87), les contrats d'interaction avec notre modèle de l'apprenant comportent quatre points :

1. **droits des acteurs** : les professeurs et tuteurs intervenant dans le cours ont des droits d'accès au modèle. Ils peuvent évaluer les compétences dont ils sont responsables et consulter l'ensemble du modèle.
2. **granularité de l'évolution** : les acteurs peuvent modifier le modèle de l'apprenant en tout temps.
3. **nombre de points de vue** : il y en a un par acteur intervenant dans le scénario, soit un par professeur ou tuteur et un pour l'apprenant (par défaut, il y a toujours un point de vue pour l'apprenant).
4. **stratégie de négociation** : c'est la stratégie 1 présentée au chapitre IV page 112 qui est utilisée (si un professeur a évalué une compétence, son évaluation est conservée, si plusieurs professeurs l'ont évalué, une moyenne est faite).

I.1.2. Scénario

Nous allons présenter dans cette section un scénario d'utilisation du contexte présenté précédemment. Ainsi, nous préciserons quels acteurs interviennent, quels acteurs sont responsables d'évaluer quelles compétences et quelles productions sont nécessaires au bon déroulement du scénario. Nous préciserons aussi comment se déroule le scénario (c'est-à-dire quelles en sont les étapes). Nous finirons en montrant les résultats de la simulation effectuée avec ce scénario. Chaque résultat permet de démontrer un aspect de notre modèle de l'apprenant comme nous l'expliquerons en exposant les résultats.

I.1.2.1. Acteurs

Deux professeurs et deux tuteurs interviennent dans ce scénario, de même qu'un apprenant, celui dont les compétences vont être évaluées. Le tableau I-2 précise les noms fictifs de ces acteurs ainsi que leur identifiant et mot de passe pour accéder au serveur SVN.

Tableau I-2. Acteurs du scénario 1

Nom	Identifiant	Mot de passe
Professeur Sylvain Robitaille	prof1	password
Professeur André Roberge	prof2	password
Tuteur Julien Contamines	tuteur1	password
Tuteur Valéry Psyché	tuteur2	password
Apprenant Lucie Moulet	etu1	password

Voyons maintenant quel acteur est responsable d'évaluer quelle compétence et à quel niveau il le fera. Le tableau suivant résume ces informations.

Tableau I-3. Compétences évaluées par différents acteurs (scénario 1)

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
Sylvain Robitaille	3		2		3	2		3			
André Roberge			4					2	4	3	
Julien Contamines		4		3	4		3				
Valéry Psyché				1					4		3
Lucie Moulet						2	4	3	3		

Nous voyons dans ce tableau que le professeur Sylvain Robitaille va évaluer la compétence C1 au niveau 3, la compétence C3 au niveau 2, et ainsi de suite.

I.1.2.2. Productions

Trois productions sont utiles au déroulement du scénario. En voici une description.

La production 1 (P1) permet d'illustrer la compétence C2 (*Planifier ses apprentissages*). C'est un tableau Excel construit par l'apprenant Lucie Moulet et contenant un échéancier avec en ligne les tâches à effectuer dans le cours et en colonne les semaines (soit le temps). Ainsi le document permet de planifier quand les tâches doivent être effectuées.

La deuxième production (P2) illustre la compétence C7 (*Synthétiser des informations de natures variées, à l'aide d'un support écrit*). C'est un fichier texte contenant un examen de synthèse de l'apprenant Lucie Moulet.

La troisième production (P3) illustre les compétences C8 (*Synthétiser ses idées en groupe en prêtant attention aux idées des autres*) et C9 (*S'adapter (auto-contrôler) aux pensées et aux opinions des autres dans un groupe*). C'est un enregistrement audio d'une réunion de groupe dans laquelle l'apprenant expose des idées qui sont débattues en groupe.

I.1.2.3. Déroulement

Le scénario comporte 12 étapes.

1. L'apprenant Lucie Moulet crée son modèle.

2. Création automatique des points de vue des acteurs intervenant (selon le contrat), soit création de cinq points de vue correspondant aux acteurs prof1, prof2, tuteur1, tuteur2 et etu1.
3. Le professeur Robitaille (prof1) évalue la compétence C1.
4. La compétence C1 est négociée.
5. Le tuteur Contamines (tuteur1) évalue la compétence C2.
L'apprenant ajoute la production P1.
L'apprenant ajoute le lien P1-C2.
Le professeur Robitaille (prof1) évalue la compétence C3.
Le professeur Roberge (prof2) évalue la compétence C3.
Le tuteur Contamines (tuteur1) évalue les compétences C4 et C5.
Le tuteur Psyché (tuteur2) évalue la compétence C4.
6. La compétence C4 est négociée.
7. Le professeur Robitaille (prof1) évalue les compétences C5 et C6.
8. La compétence C5 est négociée.
9. Le tuteur Contamines (tuteur1) certifie le lien P1-C2.
Le tuteur Contamines (tuteur1) évalue la compétence C7.
Le professeur Robitaille (prof1) évalue la compétence C8.
Le professeur Roberge (prof2) évalue les compétences C8, C9 et C10.
Le tuteur Psyché (tuteur2) évalue les compétences C9 et C11.
L'apprenant Lucie Moulet évalue les compétences C6, C7, C8 et C9.
L'apprenant Lucie Moulet ajoute P2.
L'apprenant Lucie Moulet ajoute le lien P2-C7.
Le professeur Robitaille (prof1) accrédite/tamponne P2.
Le professeur Roberge (prof2) certifie le lien P2-C7.
10. La compétence C6 est négociée.
11. La compétence C8 est négociée.
La compétence C9 est négociée.
L'apprenant Lucie Moulet ajoute P3.
L'apprenant Lucie Moulet ajoute les liens P3-C8 et P3-C9.
12. Fermeture du modèle contexte temporaire : les compétences non encore négociées le sont (les compétences C2, C3, C7, C10 et C11 sont négociées) et les points de vue (sauf celui de l'apprenant) sont supprimés.

I.1.3. Simulation

Dans cette section, nous allons montrer les résultats de la simulation effectuée dans le contexte précédent et avec le scénario présenté. Chaque section montre la validation d'un élément de notre système (ou le bon fonctionnement d'un cas de figure de la stratégie de négociation 1).

Nous avons voulu démontrer chaque cas particulier de la stratégie de négociation 1. Pour cela, quelques compétences sont négociées au milieu du scénario. Une fois toutes les possibilités démontrées, nous n'effectuons plus de négociation avant la fermeture du modèle contexte temporaire.

À chaque étape, une version du modèle de l'apprenant est créée dans SVN.

I.1.3.1. Preuve de l'ajout d'une compétence (étape 3)

Cette étape permet de montrer que la compétence C1 a été ajoutée au point de vue de l'acteur qui l'a évaluée, soit le professeur Sylvain Robitaille. La figure I-1 montre la page-écran du modèle de l'apprenant après cette étape. Nous voyons que le modèle comporte cinq points de vue, un pour chacun des acteurs, comme précisé dans le contrat et nous voyons que la compétence C1 a été ajoutée au point de vue du professeur Sylvain Robitaille. L'ontologie n'est pas précisée dans ce cas et le contexte est *cours INF1234*, nom que nous avons donné au scénario pour cette simulation.

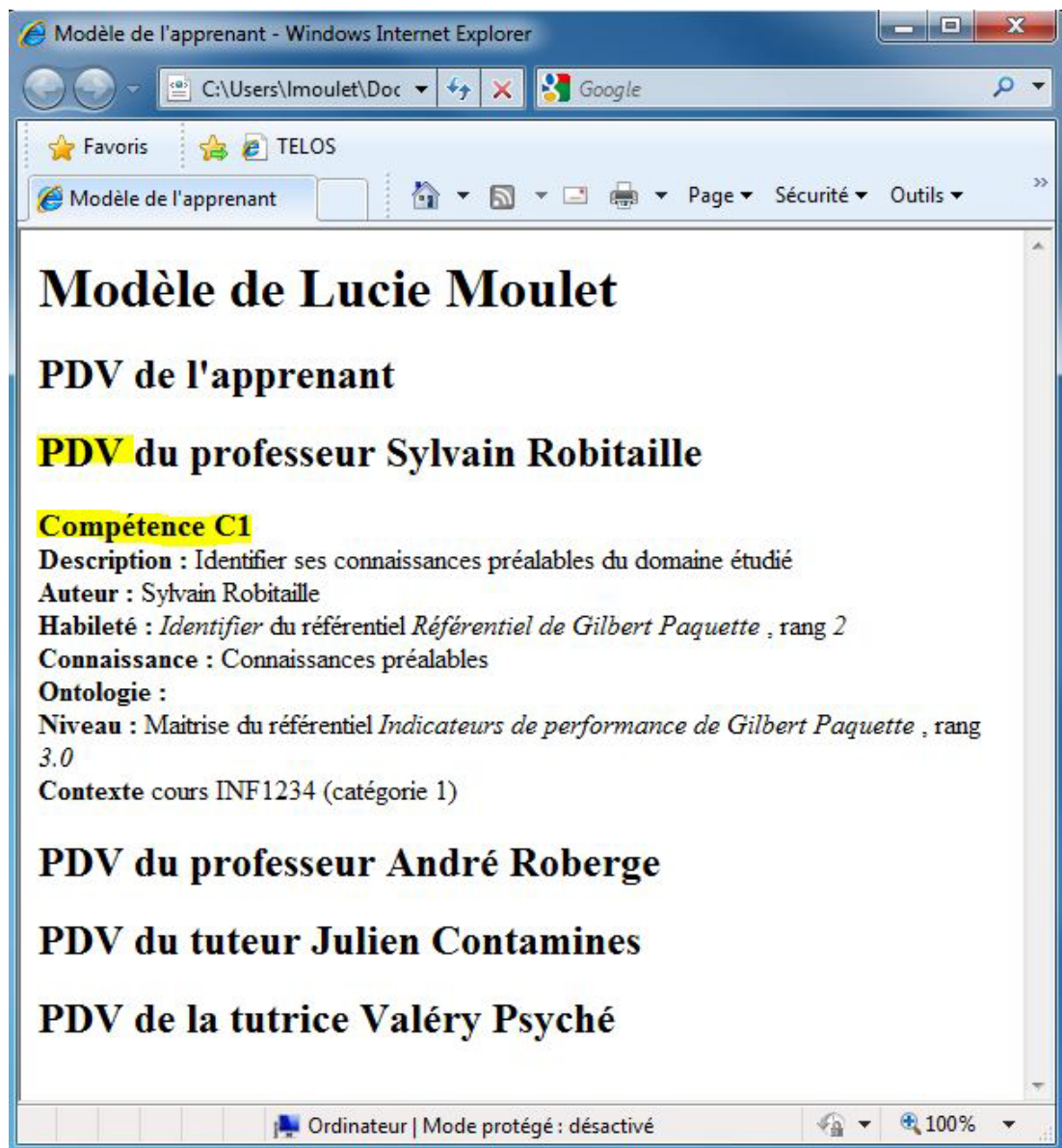


Figure I-1. Modèle de l'apprenant simulation 1 étape 3

I.1.3.2. Preuve de la stratégie de négociation (cas 1) (étape 4)

Cette étape permet de démontrer le bon fonctionnement du cas de figure 1 de la stratégie de négociation. Ce cas s'applique lorsqu'un seul acteur a évalué la compétence. Dans ce cas, la compétence est placée dans le point de vue consensus et supprimée du point de vue de l'acteur concerné. Cela est visible à la figure I-2 où la

compétence C1 n'est plus dans le point de vue du professeur Sylvain Robitaille mais dans le point de vue consensus. Notons aussi que l'auteur de la compétence est toujours le professeur Robitaille.

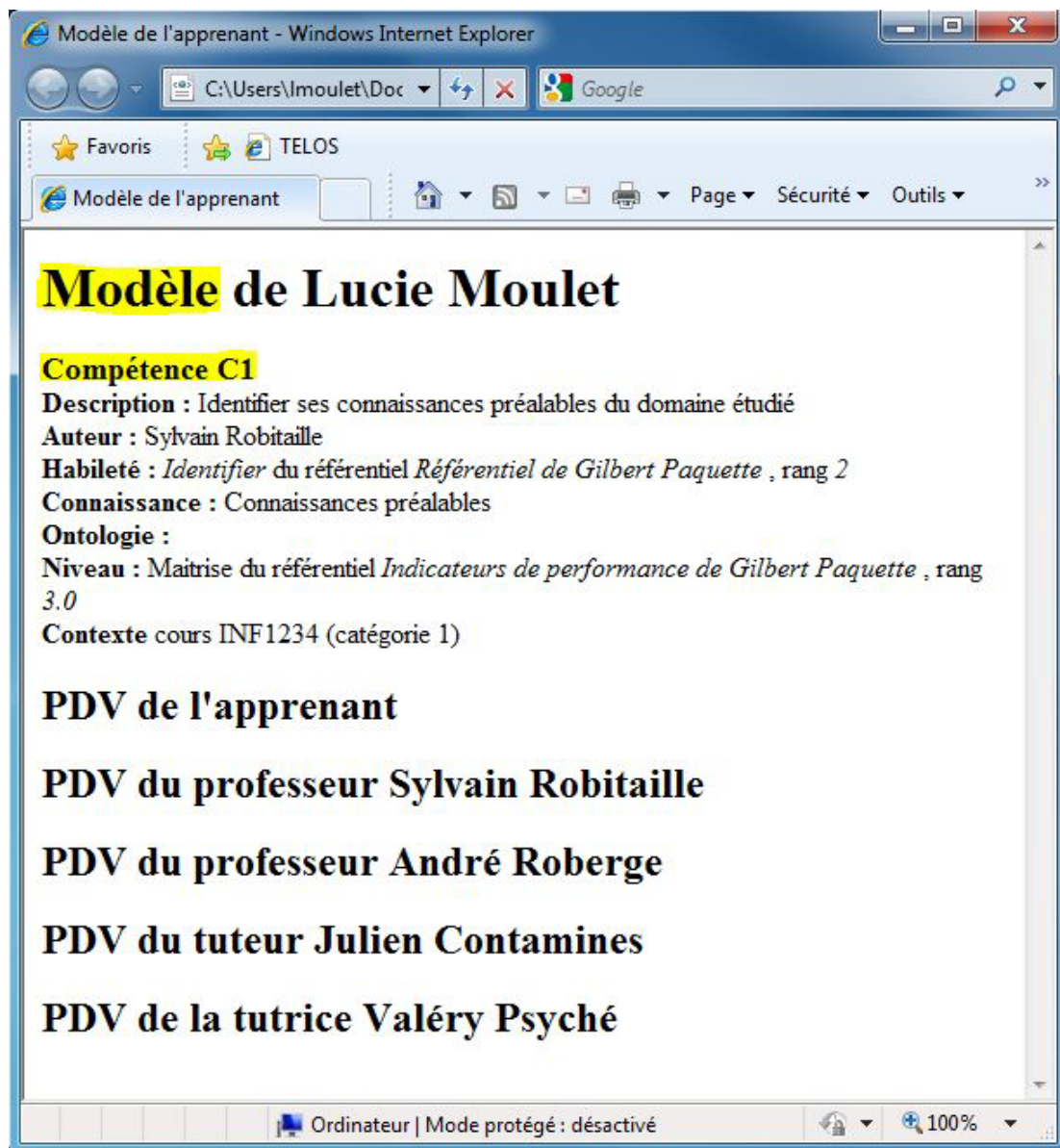


Figure I-2. Modèle de l'apprenant simulation 1 étape 4

I.1.3.3. Preuve de l'ajout d'une production et d'un lien production-compétence (étape 5)

Cette étape permet de voir qu'une production a été ajoutée dans le modèle ainsi qu'un lien entre cette production et la compétence C2. La figure I-3 montre un extrait du modèle de l'apprenant sur lequel on voit la compétence et le lien. De plus, nous voyons que des compétences ont été ajoutées dans certains points de vue (correspondant aux actions effectuées à l'étape 5 du scénario).

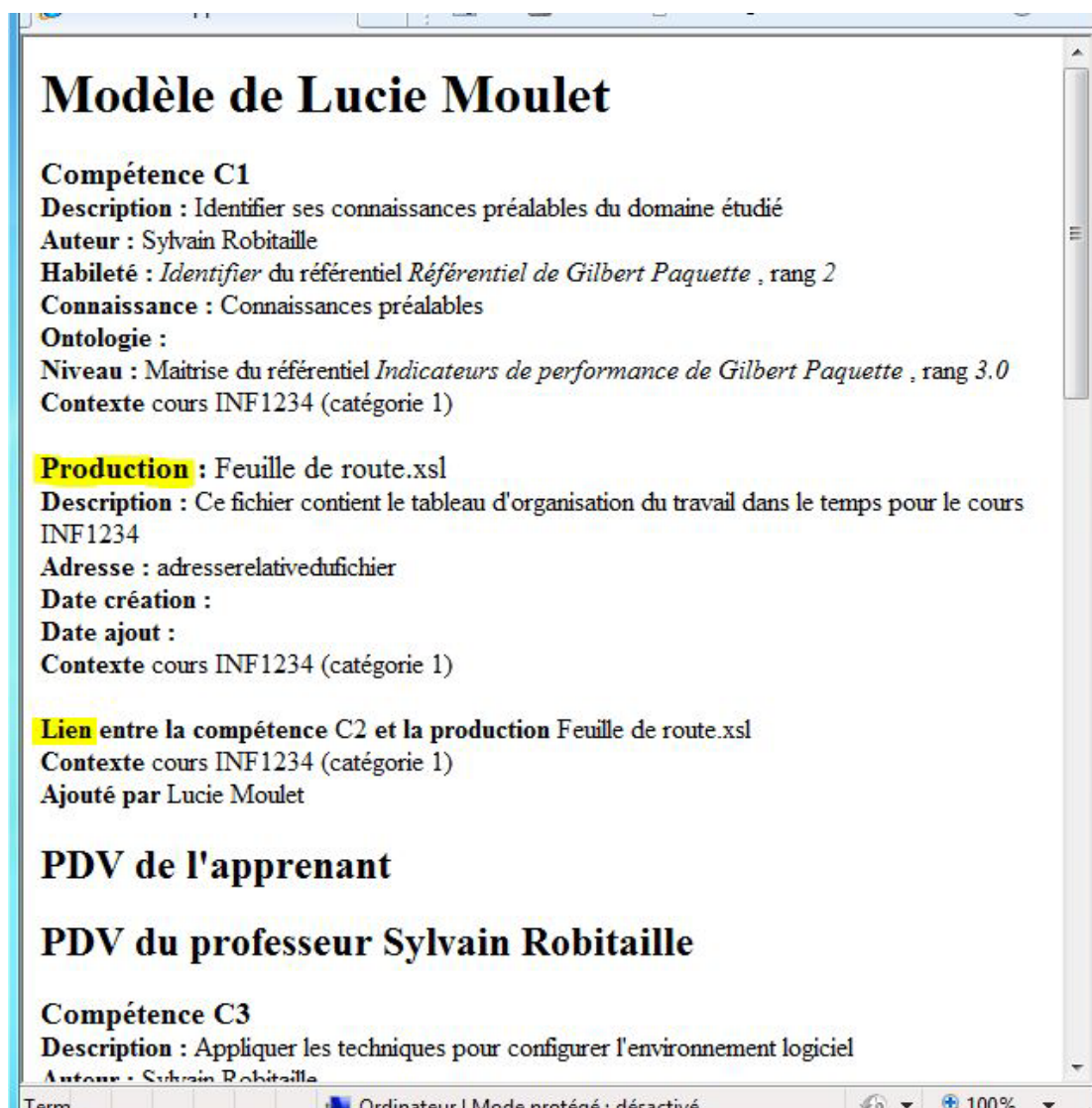


Figure I-3. Modèle de l'apprenant simulation 1 étape 5

I.1.3.4. Preuve de la stratégie de négociation (cas 2) (étape 6)

Voyons maintenant l'exécution du cas de figure 2 de la stratégie de négociation. Rappelons que dans ce cas, plusieurs acteurs du même statut ont évalué une compétence, une moyenne est donc effectuée et les noms des acteurs ayant participé à la négociation apparaissent dans le contexte de la compétence (qui est placée dans le point de vue consensus). Nous regardons particulièrement la négociation de la compétence C4, évaluée au niveau 3 par le tuteur Julien Contamines et au niveau 1 par la tutrice Valéry Psyché. La figure I-4 montre l'évaluation de la compétence C4 dans les points de vue des deux tuteurs. La figure I-5 montre le résultat de la négociation de la compétence C4. Celle-ci se situe dans le point de vue consensus avec comme valeur de niveau 2 (la moyenne des évaluations des deux tuteurs) et avec les noms des tuteurs (soit Contamines et Psyché) dans son contexte. Nous pouvons remarquer que l'auteur de la compétence C4 dans le point de vue consensus est Valéry Psyché. Rappelons que ce nom est pris au hasard parmi la liste de nom des acteurs ayant évalué la compétence au cours de la négociation.

modèle de l'apprenant

PDV du tuteur Julien Contamines

Compétence C2
Description : Planifier ses apprentissages
Auteur : Julien Contamines
Habilité : Synthétiser du référentiel *Référentiel de Gilbert Paquette*, rang 8
Connaissance : Ses apprentissages
Ontologie :
Niveau : Expertise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 4.0
Contexte cours INF1234 (catégorie 1)

Compétence C4
Description : Appliquer les techniques de communication avec le tuteur et les pairs à l'aide des outils proposés dans le cours
Auteur : Julien Contamines
Habilité : Appliquer du référentiel *Référentiel de Gilbert Paquette*, rang 5
Connaissance : Techniques de communication
Ontologie :
Niveau : Maîtrise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 3.0
Contexte cours INF1234 (catégorie 1)

Compétence C5
Description : Auto-évaluer ses apprentissages
Auteur : Julien Contamines
Habilité : Auto-controler du référentiel *Référentiel de Gilbert Paquette*, rang 10
Connaissance : Ses apprentissages
Ontologie :
Niveau : Expertise du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 4.0
Contexte cours INF1234 (catégorie 1)

PDV de la tutrice Valéry Psyché

Compétence C4
Description : Appliquer les techniques de communication avec le tuteur et les pairs à l'aide des outils proposés dans le cours
Auteur : Valéry Psyché
Habilité : Appliquer du référentiel *Référentiel de Gilbert Paquette*, rang 5
Connaissance : Techniques de communication
Ontologie :
Niveau : Sensibilisation du référentiel *Indicateurs de performance de Gilbert Paquette*, rang 1.0
Contexte cours INF1234 (catégorie 1)

Ordinateur | Mode protégé : désactivé | 100%

Figure I-4. Modèle de l'apprenant simulation 1 étape 5 (C4)

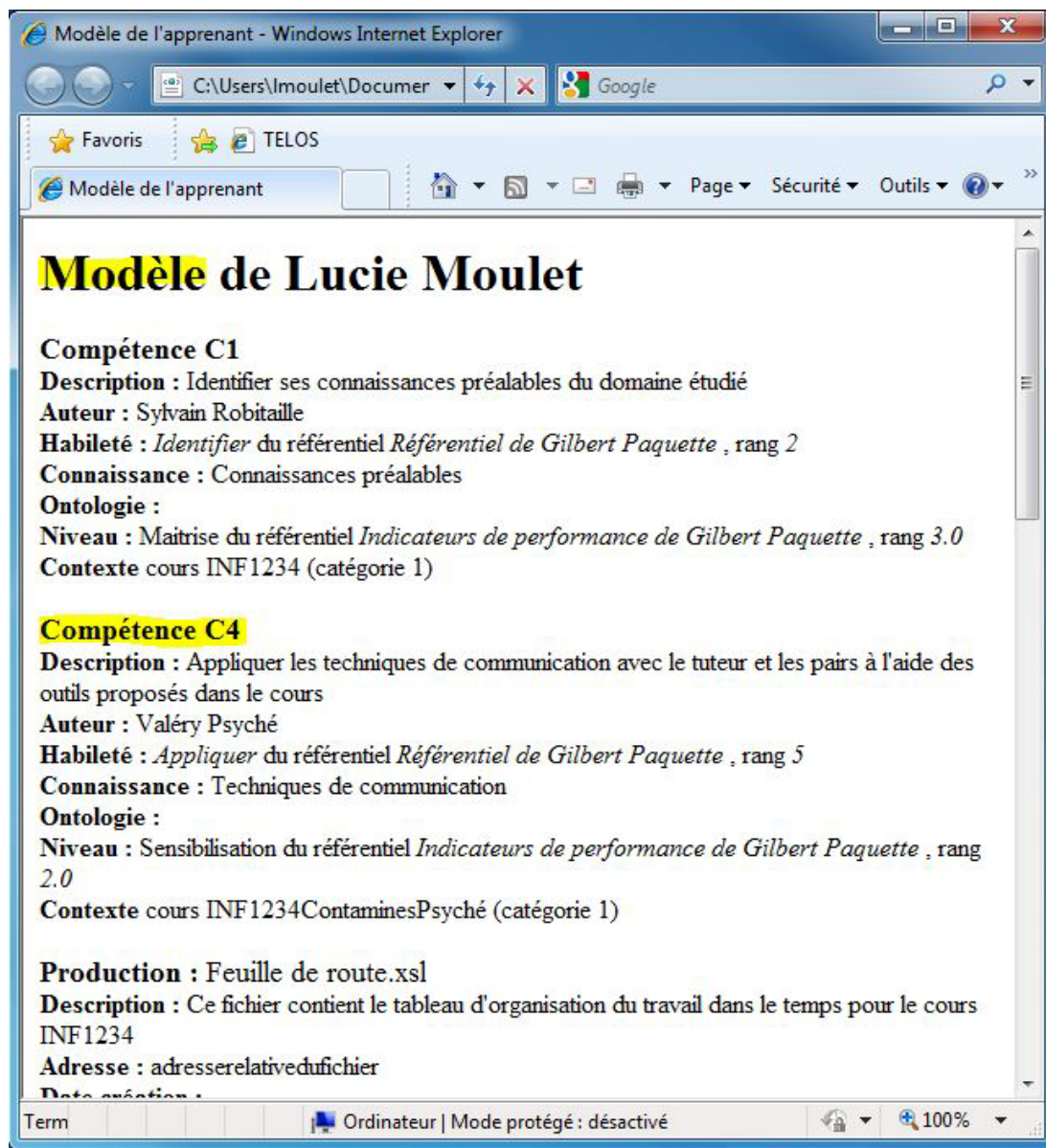


Figure I-5. Modèle de l'apprenant simulation 1 étape 6 (C4)

I.1.3.5. Preuve de la stratégie de négociation (cas 3) (étape 8)

Le cas 3 de la stratégie de négociation 1 s'applique à la compétence C5. Celle-ci a été évaluée par le professeur Sylvain Robitaille au niveau 3 et par le tuteur Julien Contamines au niveau 4. Selon la stratégie 1, c'est l'évaluation du professeur qui est conservée dans le modèle consensus. Nous montrons ici uniquement le résultat de la

négociation à la figure I-6. L'étape à laquelle la compétence C5 se trouve dans les points de vue des deux acteurs concernés est l'étape 7.

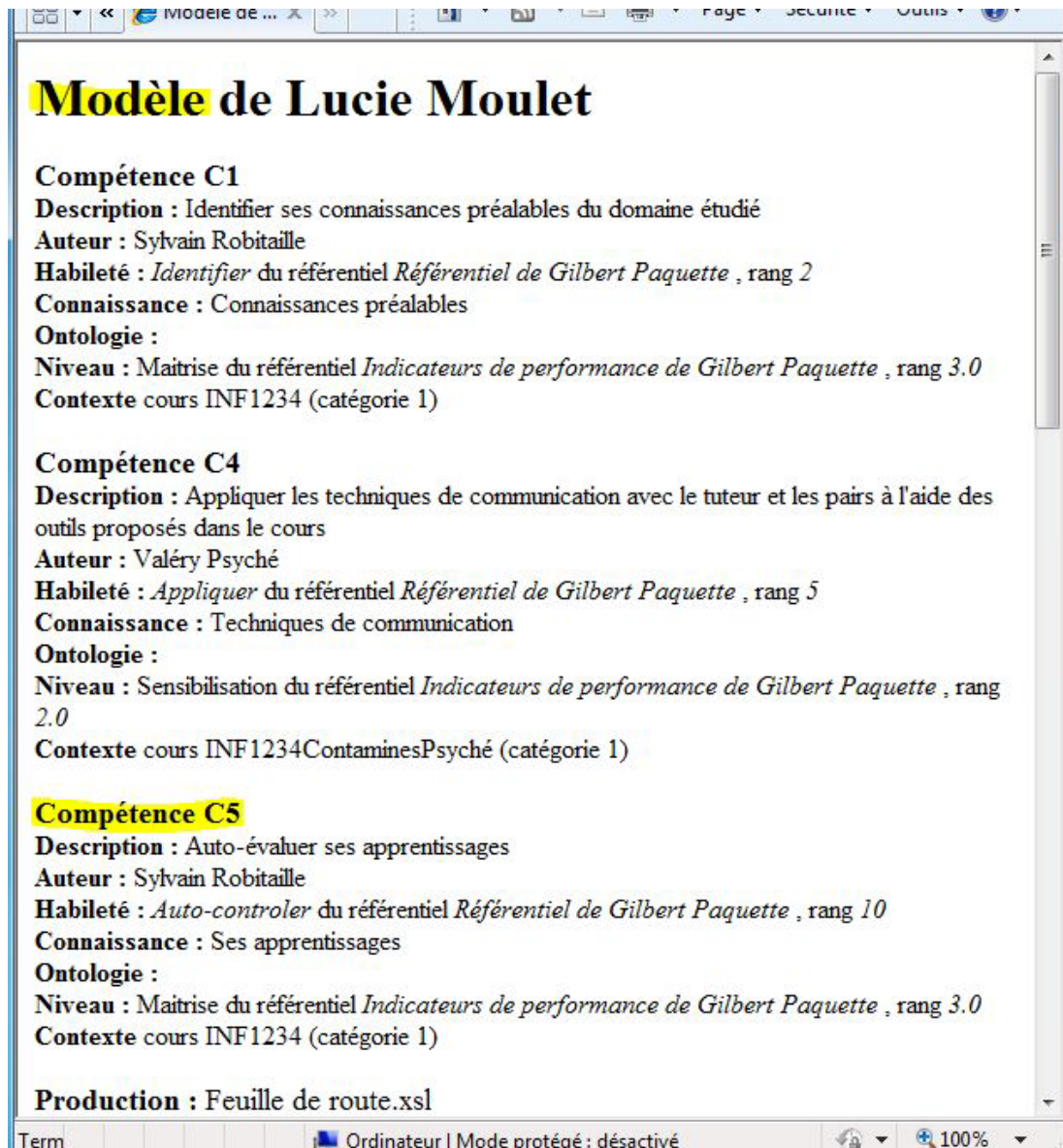


Figure I-6. Modèle de l'apprenant simulation 1 étape 8

I.1.3.6. Preuve de certification d'une production et d'un lien production-compétence (étape 9)

L'étape 9 du scénario de la simulation permet de montrer la validation d'une production et d'un lien entre une production et une compétence. Nous voyons sur la

figure I-7 que la production *Examen de synthèse* est validée par Sylvain Robitaille et que le lien entre cette production et la compétence C7 est validé par André Roberge.

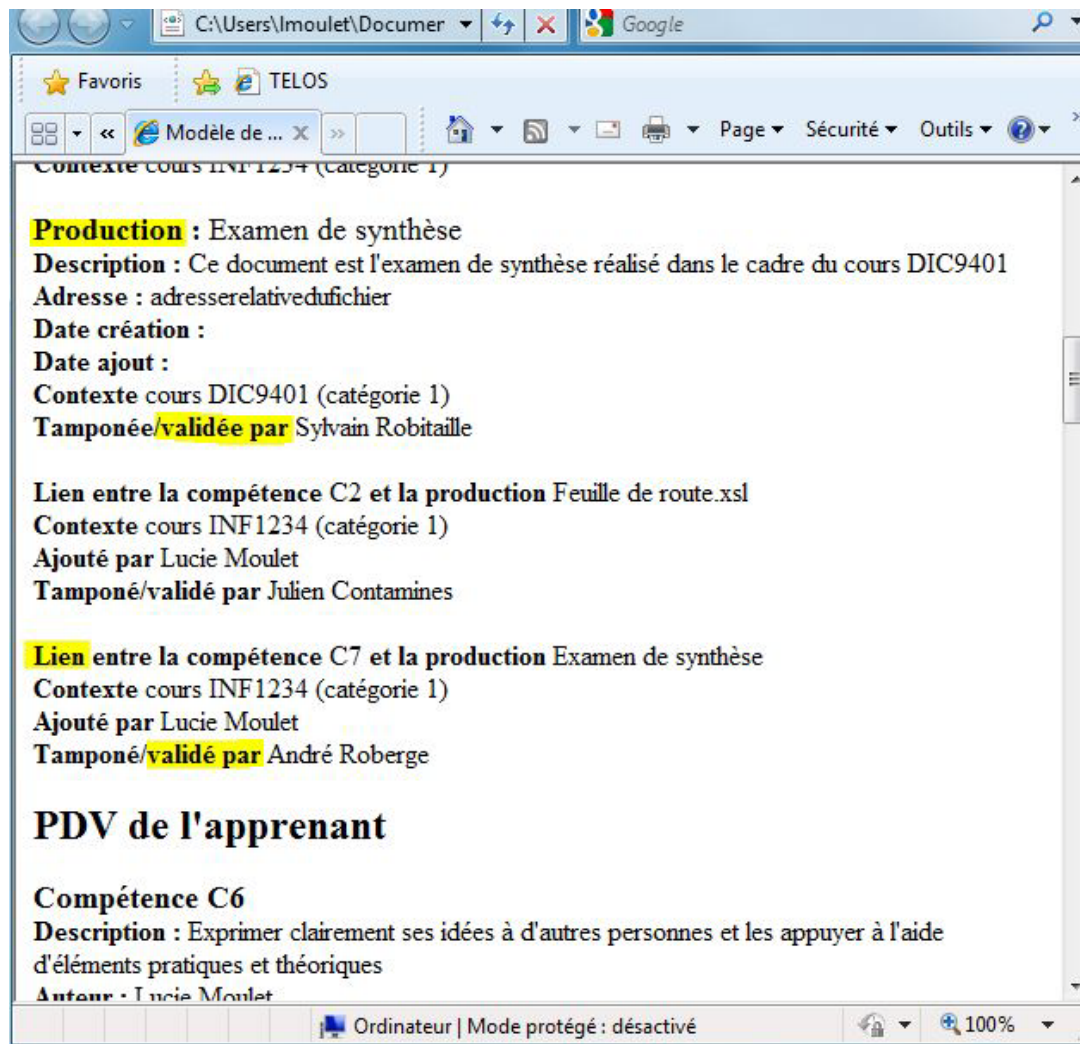


Figure I-7. Modèle de l'apprenant simulation 1 étape 9

I.1.3.7. Preuve de la stratégie de négociation (cas 4) (étape 10)

Cette étape permet de démontrer le cas de figure 4 de la stratégie de négociation, c'est-à-dire que l'apprenant et un autre type d'acteur évalue la compétence. La compétence C6 permet d'illustrer ce cas de figure. Elle est évaluée par le professeur Sylvain Robitaille au niveau 2 et l'apprenant Lucie Moulet s'auto-évalue au niveau 2. Dans ce cas de figure, lors de la négociation, la compétence est

placée dans le point de vue consensus avec le niveau donné par le professeur (et supprimé du point de vue du professeur) et elle reste dans le point de vue de l'apprenant puisque celui-ci sera conservé même à la fermeture du modèle contexte temporaire. L'évaluation de C6 dans les deux points de vue est visible à l'étape 9 et le résultat de la négociation est visible à l'étape 10. Le résultat inclut le déplacement de C6 dans le modèle consensus (voir la figure I-8), la suppression de C6 du point de vue du professeur Sylvain Robitaille et la conservation de C6 dans le point de vue de l'apprenant (voir la figure I-9).

Comment vérifier sur la figure I-8 que la compétence C6 est effectivement dans le point de vue consensus ? Tout simplement, une production est visible juste en dessous de C6 et les productions sont toujours ajoutées dans le point de vue consensus.

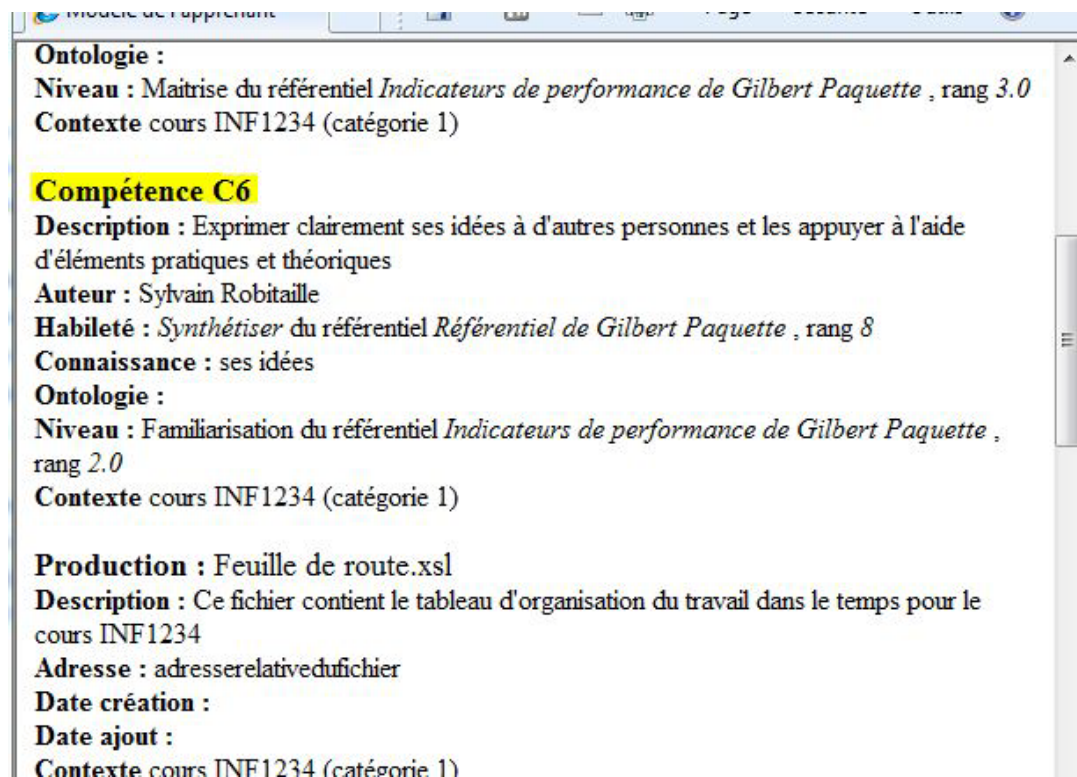


Figure I-8. Modèle de l'apprenant simulation 1 étape 10

L'ampone/valide par André Roberge

PDV de l'apprenant

Compétence C6
Description : Exprimer clairement ses idées à d'autres personnes et les appuyer à l'aide d'éléments pratiques et théoriques
Auteur : Lucie Moulet
Habileté : Synthétiser du référentiel *Référentiel de Gilbert Paquette* , rang 8
Connaissance : ses idées
Ontologie :
Niveau : Familiarisation du référentiel *Indicateurs de performance de Gilbert Paquette* , rang 2.0
Contexte cours INF1234 (catégorie 1)

Compétence C7
Description : Synthétiser des informations de natures variées, à l'aide d'un support écrit, ou à l'oral, de façon à atteindre son objectif d'information ou de communication vers des

Figure I-9. Modèle de l'apprenant simulation 1 étape 10 (point de vue de l'apprenant)

I.1.3.8. Preuve de la stratégie de négociation (cas 4 variante et cas 5) (étape 11)

Pour illustrer une variante du cas 4 de la stratégie de négociation, la compétence C8 convient parfaitement. Effectivement, celle-ci est évaluée par trois acteurs, les professeurs Sylvain Robitaille et André Roberge, aux niveaux respectifs 3 et 2, et par l'apprenant au niveau 3. Le résultat de la négociation sera que la compétence C8 sera placée dans le point de vue consensus avec un niveau de 2.5 (moyenne des évaluations des deux professeurs), elle sera supprimée des points de vue des professeurs mais conservée dans le point de vue de l'apprenant. La figure I-10 montre la compétence C8 dans le point de vue consensus.

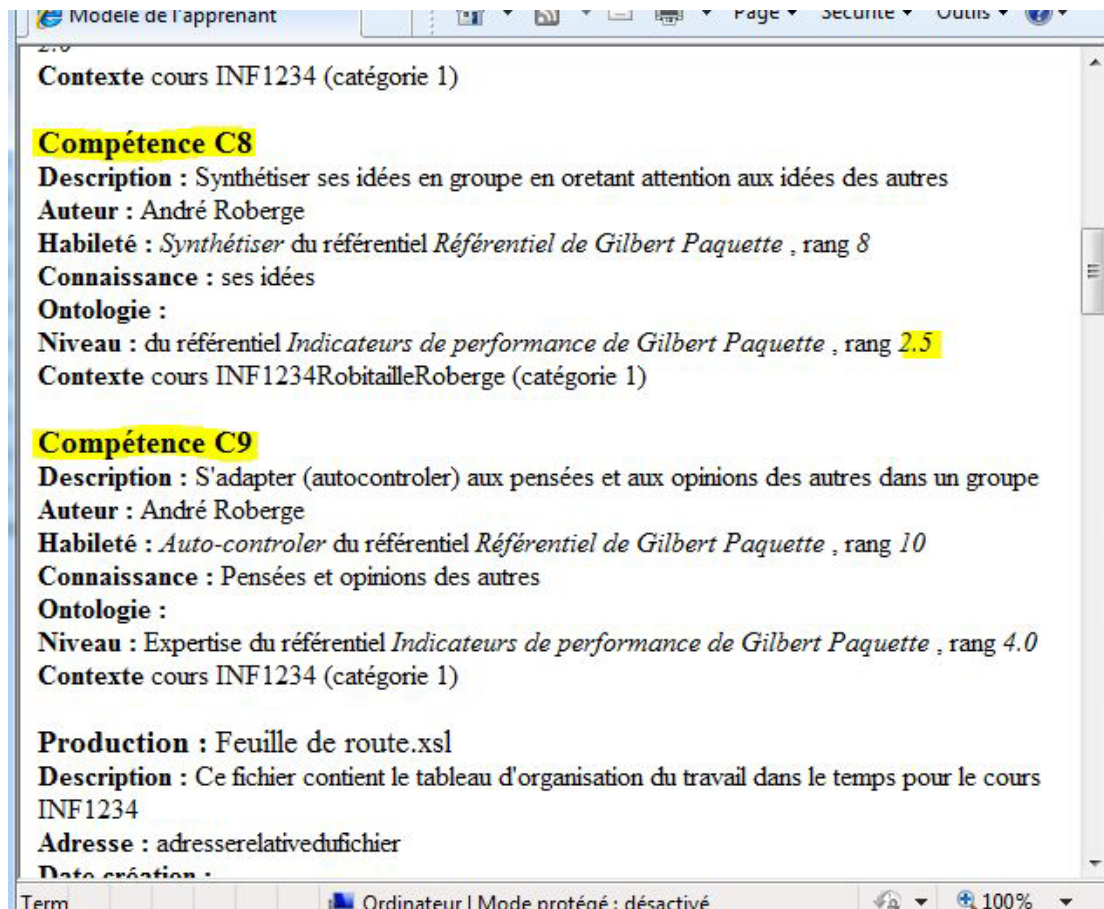


Figure I-10. Modèle de l'apprenant simulation 1 étape 11 (C8)

Pareillement, la compétence C9 démontre l'application du cas 5 de la stratégie de négociation, cas où les trois types d'acteurs évaluent la compétence. Dans notre simulation, C9 est évaluée par le professeur André Roberge au niveau 4, par le tuteur Valéry Psyché au niveau 4 et par l'apprenant au niveau 3. Comme précisé dans le chapitre IV, dans un tel cas de figure, l'évaluation du professeur est conservée dans le point de vue consensus. De plus, la compétence est supprimée des points de vue du professeur et du tuteur. Finalement, l'évaluation de l'apprenant reste dans son point de vue puisque celui-ci perdure même après la fermeture du modèle contexte temporaire. La figure I-10 montre la compétence C9 dans le point de vue consensus.

I.1.3.9. Preuve de la fermeture du modèle contexte temporaire (étape 12)

Cette étape démontre le bon fonctionnement de la fermeture du modèle contexte temporaire. En effet, les compétences non encore négociées l'ont été et se situent maintenant dans le modèle principal. Le point de vue de l'apprenant est toujours présent et tous les autres points de vue ont été supprimés.

I.2. Simulation 2 : élaboration d'un bilan de compétences

Dans cette simulation, nous nous situons dans un contexte particulier où c'est l'apprenant qui choisit de faire un bilan de compétences. Ainsi, c'est cet acteur qui pilote l'ensemble du scénario. La deuxième stratégie de négociation est utilisée, celle où l'apprenant choisit quelle évaluation il veut conserver.

Cette simulation va nous permettre de montrer une fois encore le bon fonctionnement de notre système mais plus particulièrement l'ajout d'une expérience professionnelle et d'un lien entre celle-ci et une compétence, la stratégie de négociation 2 et l'utilisation du modèle avec d'autres référentiels que ceux de Paquette.

I.2.1. Contexte

I.2.1.1. Description du contexte

Pour cette simulation, nous utilisons un contexte issu d'un texte publié en 2006 par le Ministère de l'Éducation, du Loisir et du Sport du Québec intitulé « Échelles des niveaux de compétence ». Ce document présente un référentiel de compétences et une échelle de niveau de compétence, s'apparentant à notre référentiel de niveaux. Dans ce contexte, les compétences sont présentées sous forme d'énoncés sans lien avec une connaissance identifiée dans une ontologie, ni avec une habileté (il n'y a donc pas de référentiel d'habiletés). Ce contexte peut donc être considéré comme

ayant un référencement sémantique pauvre. Toutefois, il nous permet de montrer qu'il est possible d'utiliser notre modèle de l'apprenant sans définir les compétences selon la définition de Paquette (2002a). Il a été décidé dans ce contexte d'utiliser le champ *connaissance* de la compétence pour définir le domaine, la discipline d'application de la compétence. Ceci constitue une utilisation détournée de notre modèle mais tout à fait acceptable. Un établissement scolaire peut effectivement choisir de définir ses compétences différemment de la façon envisagée dans notre proposition. Toutefois, cela réduit considérablement les possibilités d'interopérabilité ainsi que la richesse du référencement sémantique puisqu'il n'y a plus d'élément référencé sémantiquement.

De plus, même si ce référentiel de compétences cible l'enseignement au secondaire, dans le cadre de notre simulation, ce contexte est transposé au cas où l'apprenant fait un bilan de compétences. Dans ce contexte, des acteurs issus du monde scolaire vont évaluer les compétences de l'apprenant, mais c'est celui-ci qui aura le dernier mot au moment de la fermeture du modèle contexte temporaire puisqu'il choisira quelle évaluation sera conservée dans le modèle principal. Plus précisément, le contexte de cette simulation est le suivant : un apprenant fait un bilan de compétences à la fin de son secondaire. Ainsi, il demande à ses professeurs de compléter l'évaluation des compétences de leurs disciplines respectives et présentes dans le référentiel de compétences constituant le bilan. Il leur demande aussi, ainsi qu'à quelques camarades, d'évaluer des compétences transversales.

I.2.1.2. Référentiel de compétences

Nous avons ajouté au référentiel de compétences du MÉQ des compétences transversales divisées en deux disciplines : Communiquer et Prendre des décisions. Le référentiel de compétences ainsi obtenu contient 47 compétences réparties en six domaines et 14 disciplines. Les tableaux suivants présentent ce référentiel de compétences. Une colonne *Niveau* a été ajoutée à la fin, elle sera présentée ultérieurement mais est placée ici afin d'éviter de répéter ces tableaux.

Tableau I-4. Référentiel de compétences du contexte 2 (MÉQ) – 1/2

Domaine	Discipline	C#	Énoncé	Niv.
Domaine des langues	Français	F1	Lire et apprécier des textes variés	3
		F2	Écrire des textes variés	4
		F3	Communiquer oralement selon des modalités variées	3
	English as a Second Language, Core Program	E1	Interacts orally in English	3
		E2	Reinvests understanding of texts	3
		E3	Writes and produces texts	3
Domaine de la mathématique, de la science et de la technologie	Mathématique	M1	Résoudre une situation-problème	5
		M2	Déployer un raisonnement mathématique	4
		M3	Communiquer à l'aide du langage mathématique	4
	Science et technologie	S1	Chercher des réponses ou des solutions à des problèmes d'ordre scientifique ou technologique	4
		S2	Mettre à profit ses connaissances scientifiques et technologiques	3
		S3	Communiquer à l'aide des langages utilisés en science et technologie	2
Domaine de l'univers social	Géographie	G1	Lire l'organisation d'un territoire	4
		G2	Interpréter un enjeu territorial	3
		G3	Construire sa conscience citoyenne à l'échelle planétaire	4
	Histoire et éducation à la citoyenneté	H1	Interroger les réalités sociales dans une perspective historique	4
		H2	Interpréter les réalités sociales à l'aide de la méthode historique	5
		H3	Construire sa conscience citoyenne à l'aide de l'histoire	5

Tableau I-5. Référentiel de compétences du contexte 2 (MÉQ) - 2/2

Domaine	Discipline	C#	Énoncé	Niv.
Domaine des arts	Art dramatique	A1	Créer des œuvres dramatiques	3
		A2	Interpréter des œuvres dramatiques	4
		A3	Apprécier des œuvres dramatiques	3
	Arts plastiques	AP1	Créer des images personnelles	3
		AP2	Créer des images médiatiques	2
		AP3	Apprécier des œuvres d'art et des objets culturels du patrimoine artistique, des images personnelles et des images médiatiques	3
	Danse	D1	Créer des danses	3
		D2	Interpréter des danses	3
		D3	Apprécier des danses	3
	Musique	MU1	Créer des œuvres musicales	3
		MU2	Interpréter des œuvres musicales	2
		MU3	Apprécier des œuvres musicales	2
Domaine du développement personnel	Éducation physique et à la santé	EP1	Agir dans divers contextes de pratique d'activités physiques	4
		EP2	Interagir dans divers contextes de pratique d'activités physiques	4
		EP3	Adopter un mode de vie sain et actif	5
	Enseignement moral	EM1	Se positionner, de façon réfléchie, au regard d'enjeux d'ordre éthique	4
		EM2	Construire un référentiel moral	4
		EM3	Pratiquer le dialogue moral	4
Domaine transversal	Communiquer	C1	Comprendre des rôles	
		C2	Comprendre des contextes	
		C3	Comprendre des intentions	
		C4	Comprendre des messages	
		C5	Formuler des messages	
		C6	Produire une œuvre personnelle	
	Prendre des décisions	C7	Utiliser l'information	
		C8	Fixer des objectifs	
		C9	Faire un plan	
		C10	Résoudre des problèmes	
		C11	Réaliser des projets	

I.2.1.3. Référentiel de niveaux

Le référentiel de niveaux utilisé dans ce contexte est différent de celui de Paquette (2002a). C'est le référentiel proposé par le Ministère de l'Éducation du

Québec et qui est présenté dans le tableau I-6. Il contient cinq niveaux permettant d'évaluer les compétences.

Tableau I-6. Référentiel de niveaux du contexte 2 (MÉQ)

Niveau	Compétence	Jugement global en fin de cycle
1	Très peu développée	La compétence de l'élève est nettement en deçà des exigences
2	Peu développée	La compétence de l'élève est en deçà des exigences
3	Acceptable	La compétence de l'élève satisfait minimalement aux exigences
4	Assurée	La compétence de l'élève satisfait clairement aux exigences
5	Marquée	La compétence de l'élève dépasse les exigences

La première colonne *Niveau* est le rang, la seconde colonne *Compétence* définit le niveau, elle correspond au nom du niveau dans notre système. Par exemple, pour le niveau 1, la compétence est *très peu développée* par l'étudiant. La dernière colonne représente une description du niveau.

I.2.1.4. Contrat

Voici les valeurs des quatre points constituant un contrat :

5. **droits des acteurs** : les professeurs et tuteurs peuvent évaluer les compétences des disciplines dont ils sont responsables. De plus, les acteurs ayant reçu une invitation de l'apprenant peuvent évaluer ses compétences transversales.
6. **granularité de l'évolution** : les acteurs peuvent modifier le modèle de l'apprenant en tout temps.
7. **nombre de points de vue** : il y en a un par acteurs ayant reçu une invitation de l'apprenant à évaluer ses compétences.
8. **stratégie de négociation** : c'est la stratégie 2 présentée au chapitre IV page 113 qui est utilisée (c'est-à-dire que si plusieurs acteurs ont évalué une compétence, les différentes évaluations sont listées et l'apprenant choisit celle qu'il veut conserver).

I.2.2. Scénario

Le contexte de ce scénario est du type 2, *référentiel de compétences libre* et il est appelé *Bilan de l'apprenant*.

I.2.2.1. Acteurs

Sept professeurs, quatre tuteurs et trois apprenants interviennent dans ce scénario. Ainsi 14 points de vue sont créés. Le tableau I-7 précise les noms fictifs de ces acteurs, leur discipline ainsi que leur identifiant et mot de passe pour accéder au serveur SVN. Ugo Moulet est l'apprenant dont c'est le modèle.

Tableau I-7. Acteurs du scénario 2

Nom	Discipline	Identifiant	Mot de passe
Professeur Xuan Decroocq	Français (langue d'enseignement)	prof3	password
Professeur Martine Peigner	Mathématique	prof4	password
Professeur Annie Beauchamp	Science et technologie	prof5	password
Professeur Aurélie Grillot	Géographie, Histoire et éducation à la citoyenneté	prof6	password
Professeur Loïc Bélingard	Arts plastiques	prof7	password
Professeur Nicolas Lecomte	Danse	prof8	password
Professeur Noémie Poulain	Musique	prof9	password
Tuteur Robert Rothman	English as a Second Language, Core Program	tuteur3	password
Tuteur Délia Rogozan	Art dramatique	tuteur4	password
Tuteur Alexis Miara	Éducation physique et à la santé	tuteur5	password
Tuteur Igor Manceau	Enseignement moral	tuteur6	password
Apprenant Ugo Moulet	/	etu2	password
Apprenant Laurence Leclercq	/	etu3	password
Apprenant Chloé Buffet	/	etu4	password

Dans ce scénario, les professeurs et tuteurs évaluent les compétences correspondant à leur discipline d'enseignement. Pour savoir à quel niveau les compétences disciplinaires sont évaluées, il suffit de se reporter à la colonne *Niv.* (représentant le niveau) des tableaux I-4 et I-5.

Concernant les compétences transversales, quatre professeurs et deux apprenants participent à leur évaluation après l'invitation de l'apprenant Ugo Moulet : les professeurs de français, de mathématique, d'éducation physique et à la santé et celui d'enseignement moral, ainsi que les étudiantes Laurence Leclercq et Chloé Buffet. Le tableau suivant présente le niveau de performance accordé par chaque professeur pour les six compétences de la discipline *Communiquer* et les cinq compétences de la discipline *Prendre des décisions*. Les étudiantes n'évaluent que les compétences de *Communiquer*.

Tableau I-8. Évaluation des compétences transversales

	Communiquer						Prendre des décisions				
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
Professeur de français Xuan Decroocq	3	4	3	2	4	4	2	3	3	4	5
Professeur de mathématique Martine Peigner	3	3	2	2	3	4	1	4	3	5	5
Tuteur d'éducation physique et à la santé Alexis Miara	2	4	4	3	4	3	2	3	2	4	4
Tuteur d'enseignement moral Igor Manceau	4	3	3	4	4	2	3	3	2	3	5
Laurence Leclercq	5	3	4	3	4	3	/	/	/	/	/
Chloé Buffet	3	4	3	3	3	4	/	/	/	/	/

I.2.2.2. Productions

Deux productions sont utilisées dans ce scénario. La première, P1, illustre la compétence F2, *Écrire des textes variés*. C'est un fichier texte intitulé *Recueil de textes.docx* et regroupant un ensemble de textes écrits par l'apprenant. La seconde production (P2) est un fichier *Power Point* intitulé *PubLessive.ppt*. Il contient une image créée par l'apprenant. Cette production illustre la compétence AP2, *Créer des images médiatiques*.

I.2.2.3. Expériences professionnelles

Une expérience professionnelle (EP1) est utilisée au cours du scénario. C'est un stage effectué entre le 3 mai 2008 et le 10 septembre 2008 par l'apprenant au Théâtre du rideau. Au cours de cette expérience, l'apprenant concerné a joué un rôle secondaire dans une pièce présentée pendant l'été. Cette expérience professionnelle illustre la compétence A2, *Interpréter des œuvres dramatiques*.

I.2.2.4. Déroulement

Le scénario comporte 8 étapes.

1. L'apprenant Ugo Moulet crée son modèle.
2. Le modèle est rempli, c'est-à-dire que les professeurs, tuteurs et apprenants évaluent les différentes compétences (tel que précisé dans les paragraphes précédents).
3. L'apprenant Ugo Moulet ajoute l'expérience professionnelle EP1.
L'apprenant Ugo Moulet ajoute un lien entre l'expérience professionnelle EP1 et la compétence A2.
4. L'apprenant Ugo Moulet ajoute les productions P1 et P2.
L'apprenant Ugo Moulet ajoute un lien entre la production P1 et la compétence F2.
L'apprenant Ugo Moulet ajoute un lien entre la production P2 et la compétence AP2.
Le lien P1-F2 est certifié par le professeur Xuan Decroocq.
Le lien P2-AP2 est certifié par le professeur Loïc Bélingard.
5. La compétence F1 est négociée.
6. Toutes les compétences évaluées par un seul acteur sont négociées (c'est-à-dire toutes les compétences autres que les compétences transversales).
7. La compétence C1 est négociée.
8. Fermeture du modèle contexte temporaire. Les compétences non encore négociées le sont (soit C2 à C11, les compétences transversales), tous les points de vue sauf celui de l'apprenant sont supprimés.

I.2.3. Simulation

Le modèle de l'apprenant Ugo Moulet est modifié au fur et à mesure du déroulement du scénario. L'ensemble des versions du modèle (selon les différentes étapes de déroulement) est disponible à l'appendice H.

I.2.3.1. Preuve de l'utilisation de différents référentiels (étape 2)

Nous voulons montrer ici l'utilisation de notre modèle de l'apprenant avec d'autres référentiels que ceux de Paquette. Plus précisément, dans cette simulation, le référentiel de compétences utilisé n'est pas fondé sur la définition de compétence de Paquette et ne comporte ainsi pas de référentiel d'habiletés. Aussi, le référentiel de niveaux utilisé est celui du MÉQ.

Nous voyons sur la figure I-11 que les compétences n'ont pas d'habileté. De plus, le champ *connaissance* est utilisé, comme dit précédemment, pour préciser la discipline de la compétence, et le champ ontologie est utilisé pour préciser l'origine du référentiel de compétences. Sur la figure, l'intitulé de ces champs est d'ailleurs effectivement *Discipline* et *Origine*. En effet, chaque institution peut définir son propre fichier d'affichage des modèles en XML. Par contre, comme nous l'avons déjà précisé, cela réduit et même détruit les possibilités d'interopérabilité et de référencement sémantique. D'interopérabilité puisqu'une autre institution ne pourra pas interpréter ces compétences étant donné que le modèle est utilisé de façon détournée. Comment savoir que le champ connaissance contient en fait la discipline ? Et de référencement sémantique puisque les compétences ne sont plus liées ni à un référentiel d'habiletés, ni à une ontologie.

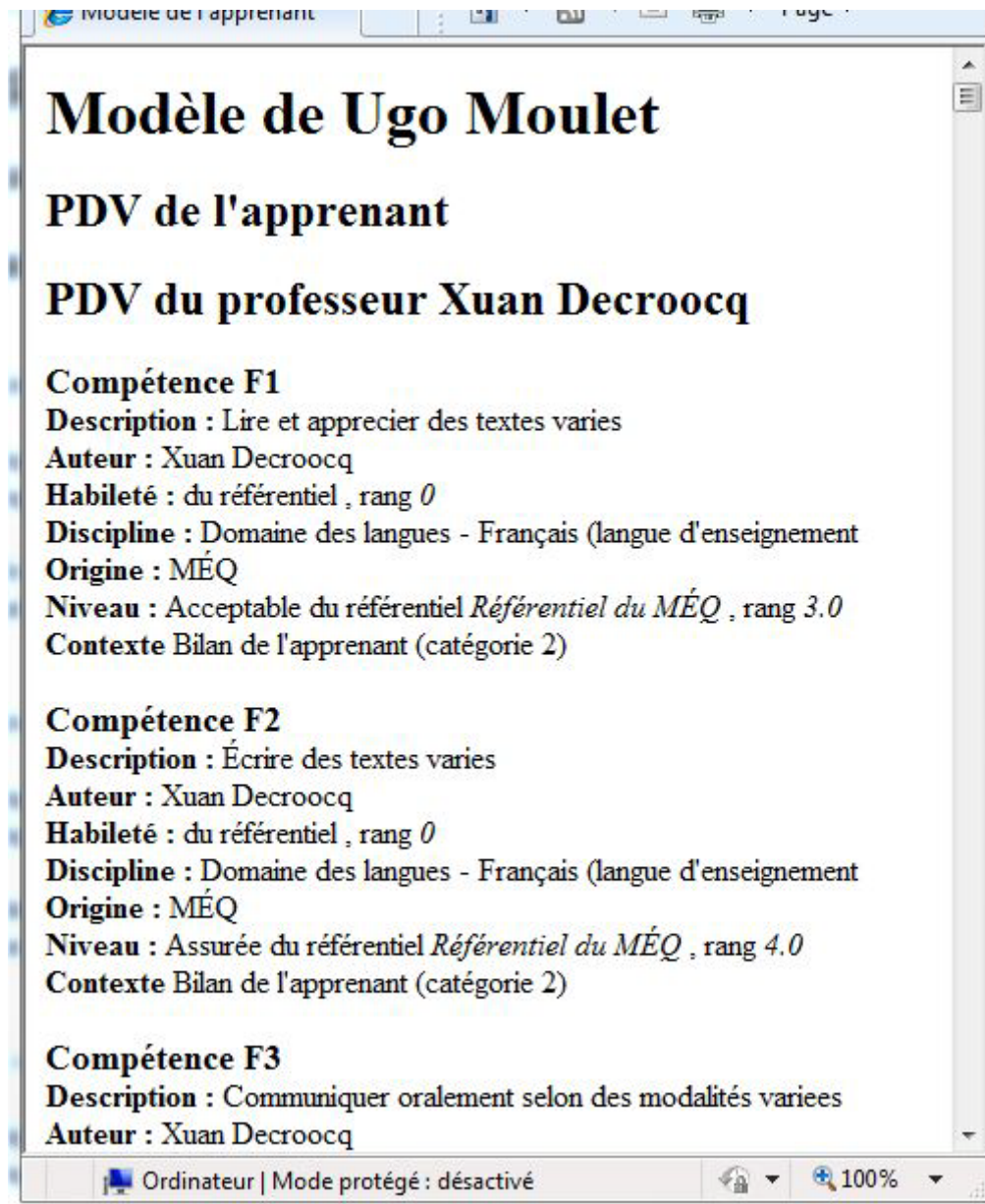


Figure I-11. Modèle de l'apprenant simulation 2 étape 2

I.2.3.2. Preuve de l'ajout d'une expérience et d'un lien entre une expérience professionnelle et une compétence (étape 3)

Cette étape montre le bon fonctionnement de la fonction d'ajout d'une expérience professionnelle ainsi que celle d'ajout d'un lien entre une expérience

professionnelle et une compétence. La figure I-12 montre en effet la présence de l'expérience professionnelle *Stage acteur* ainsi qu'un lien entre celle-ci et la compétence A2.

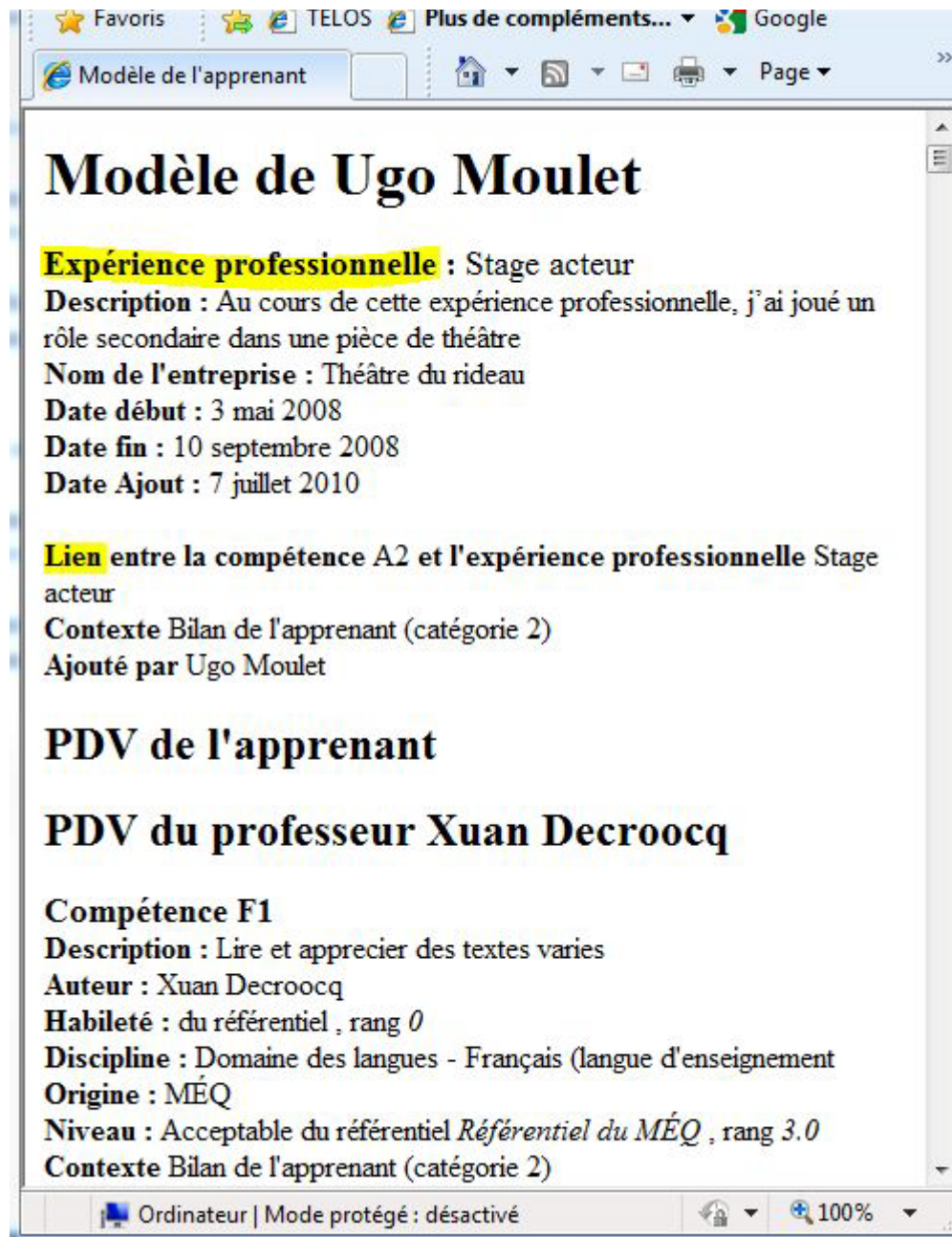


Figure I-12. Modèle de l'apprenant simulation 2 étape 3

I.2.3.3. Preuve de la stratégie de négociation (cas 1) (étape 5)

La deuxième stratégie de négociation a été présentée dans le chapitre IV page 113. Le premier cas de figure est celui où la compétence a été évaluée par un seul acteur. Dans ce cas, à la négociation, la compétence est placée dans le modèle consensus et supprimée du point de vue de l'acteur qui l'a évaluée. Nous illustrons ce cas avec la compétence F1, *Lire et apprécier des textes variés*, qui a été évaluée par le professeur Xuan Decroocq. La figure I-13 nous montre la compétence après la négociation. Nous pouvons voir qu'elle est dans le modèle principal avec pour auteur Xuan Decroocq.

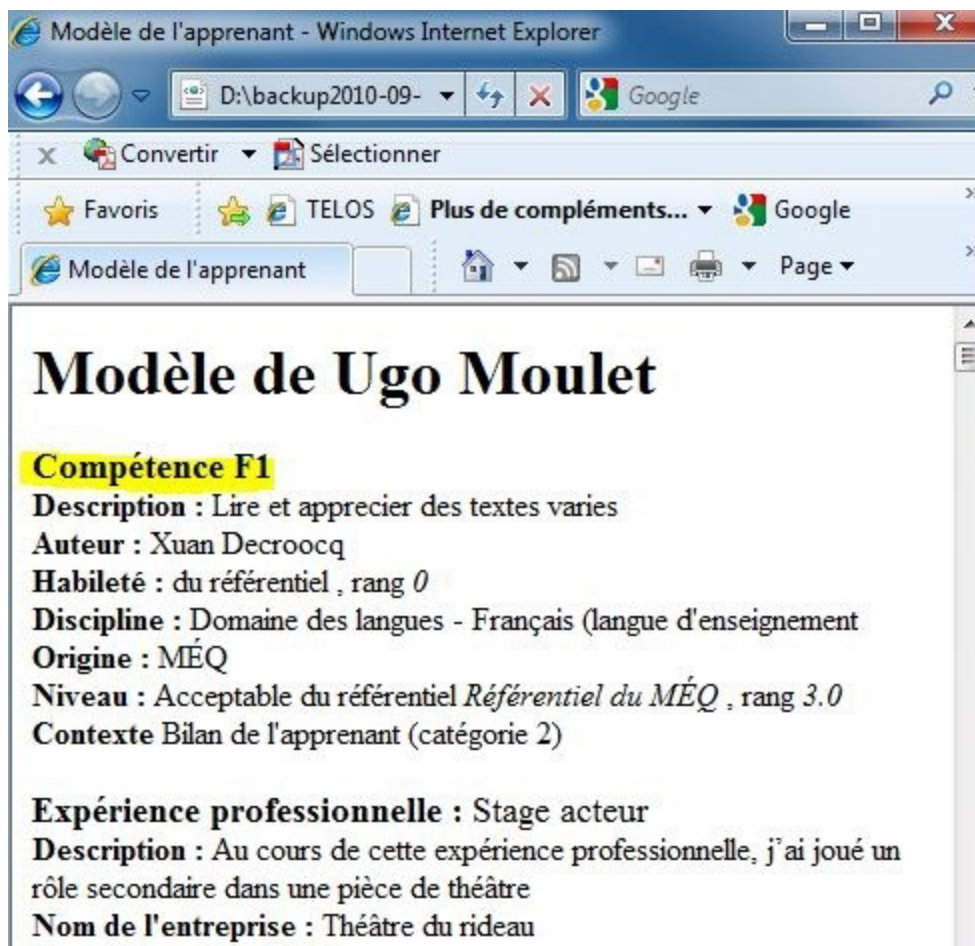


Figure I-13. Modèle de l'apprenant simulation 2 étape 5

I.2.3.4. Preuve de la stratégie de négociation (cas 2) (étape 7)

Nous montrons à cette étape le bon fonctionnement du deuxième cas de figure de la stratégie de négociation 2. Dans ce cas, le système affiche la liste des évaluations de la compétence à l'écran et l'apprenant sélectionne celle qu'il souhaite conserver dans le modèle consensus. La figure I-14 montre la liste affichée et le texte tapé par l'apprenant en réponse. L'interface utilisée est celle du logiciel permettant de traiter des fichiers Java (soit IntelliJ IDEA) car nous n'avons pas développé d'interface utilisateur dans notre prototype. Le texte apparaissant en vert est ce qui a été entré par l'utilisateur.

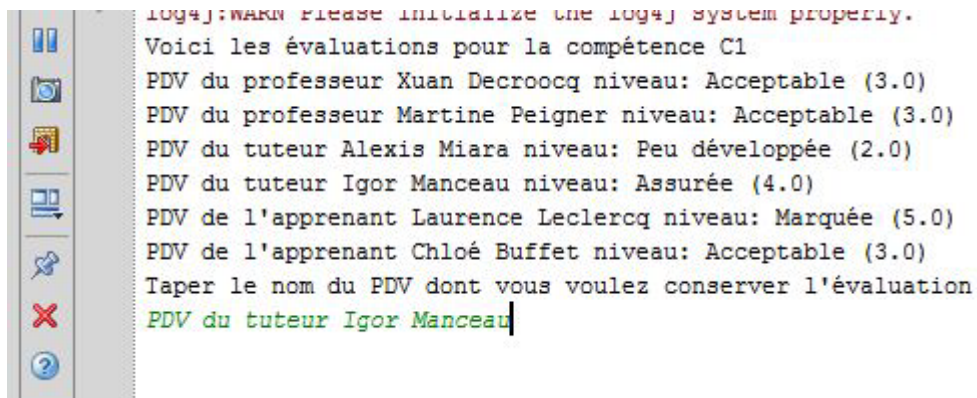


Figure I-14. Choix de l'évaluation pour C1

La figure I-15 montre que la compétence C1 dans le point de vue consensus a bien comme auteur *Igor Manceau* comme choisi par l'apprenant.

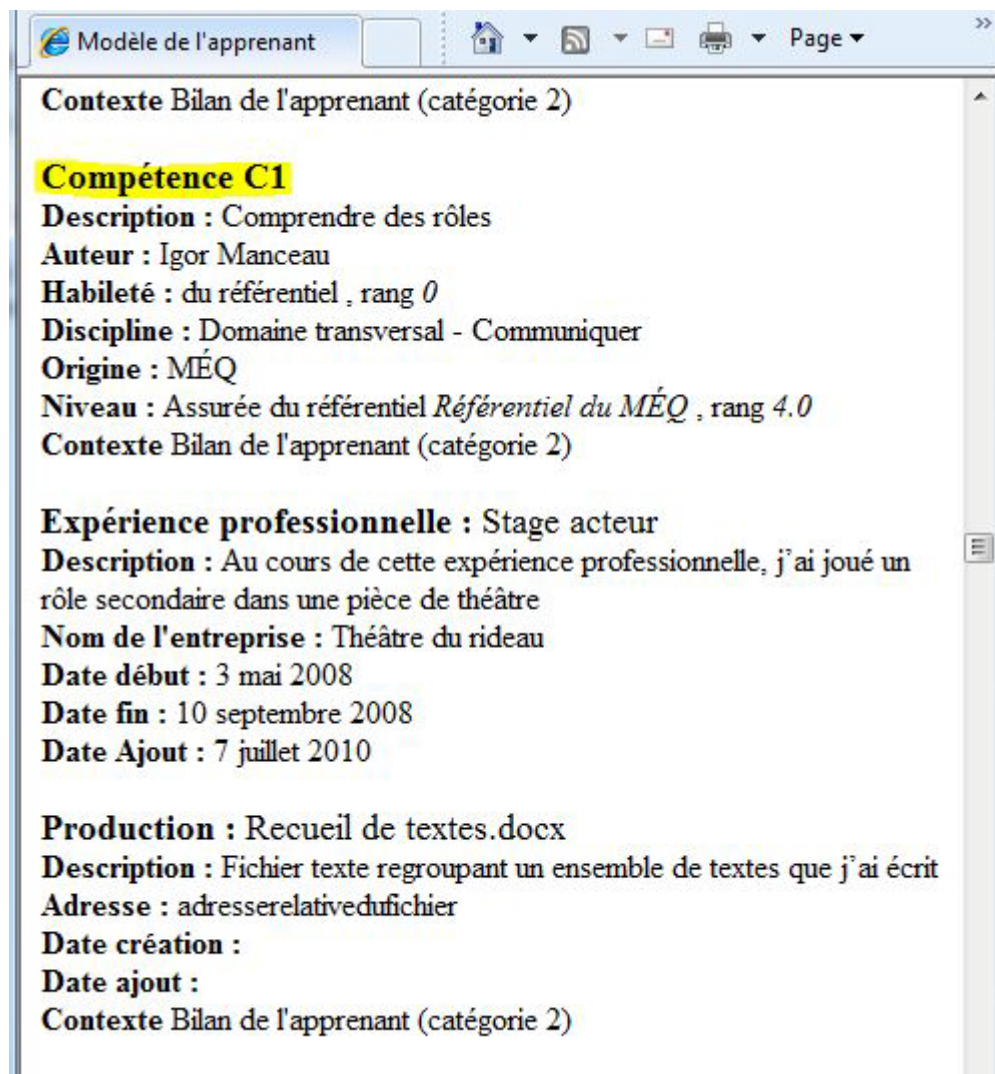


Figure I-15. Modèle de l'apprenant simulation 2 étape 7

I.3. Simulation 3 : Système d'apprentissage en ligne

Cette troisième et dernière simulation vise à démontrer le bon fonctionnement du lien de notre système avec un système d'apprentissage en ligne, TELOS. Dans cette simulation, nous sommes dans un contexte de cours en ligne « automatique » c'est-à-dire que c'est TELOS qui ajoute des compétences dans le modèle de l'apprenant.

Plus précisément, cette simulation vise à faire la preuve de l'utilisation d'ontologies et de la bonne interaction entre notre système Nosma et un autre système, en l'occurrence TELOS; interaction qui se fera dans les deux sens, soit ajout d'information dans Nosma et demande d'information de Nosma.

I.3.1. Contexte

I.3.1.1. Description du contexte

Le contexte de cette simulation est inspiré du cours DIC 9330 *Modélisation cognitive et design pédagogique*, suivi dans le cadre du programme de doctorat en informatique cognitive. Un des travaux de ce cours consistait à construire un scénario pédagogique visant l'obtention de compétences portant sur le repérage d'information et tirées du profil de compétences du jeune chercheur nommé *Joint Skills Statement* adopté par l'*UK Research Councils* (Research councils Uk, 2005).

Dans cette simulation, il n'y a pas de point de vue car le scénario correspond à l'interaction de l'apprenant avec un système, TELOS. C'est ce système qui ajoute les compétences et uniquement lui, ainsi aucune négociation n'est requise. Lorsque l'apprenant termine une activité du scénario pédagogique, TELOS ajoute automatiquement une compétence dans son modèle. En réalité, une évaluation devrait probablement être faite pour s'assurer que l'apprenant possède effectivement la compétence ajoutée (que ce soit par un test automatique du type QCM ou par un acteur humain). Toutefois, dans le cadre de notre thèse et pour mener à bien notre simulation, nous considérons que TELOS peut ajouter une compétence lorsque l'apprenant termine une activité.

I.3.1.2. Référentiel de compétences

Nous avons adapté le référentiel de compétences du *UK Research Councils* pour que celui-ci soit compatible avec la définition de compétence de Paquette, c'est-

à-dire que les compétences soient décrites par une connaissance (en lien avec une ontologie) et une habileté. Le tableau I-9 présente le référentiel de compétences (modifié) de ce contexte. Il est constitué de huit compétences identifiées selon un énoncé, une connaissance, une habileté (et son rang) ainsi que le niveau auquel la compétence sera évaluée au cours du scénario pédagogique. Les référentiels d'habiletés et de niveaux sont ceux de Paquette (2002a), présentés au chapitre I page 32.

Tableau I-9. Référentiel de compétences du contexte 3

C	Énoncé de la compétence	Connaissance	Habileté	Niveau
C1	Identifier les sources d'informations et documents, incluant le degré de confiance relatif à ces sources	Sources d'informations (ontologie)	Intégrer (2)	4
C2	Identifier les types d'informations et documents, leur statut relatif à la publication	Types d'information (ontologie)	Intégrer (2)	4
C3	Connaitre et utiliser les moyens et protocoles d'accès	Protocoles d'accès	Appliquer (5)	3
C4	Identifier les personnes-ressources pertinentes (conseil)	Personnes-ressources (ontologie)	Intégrer (2)	2
C5	Élaborer des stratégies de repérage, incluant correction et raffinement des requêtes	Stratégies de repérage (ontologie)	Synthétiser (8)	2
C6	Identifier les outils disponibles pour les différentes étapes du repérage, et pour préparer les étapes suivantes (récupération, création de bibliographies, de notes de lecture)	Outils	Intégrer (2)	1
C7	Respecter les règles d'éthique et les conventions du milieu	Règles d'éthique	Appliquer (5)	3
C8	Développer une position personnelle sur les pratiques de collaboration entre futurs chercheurs relativement à ce repérage	Position personnelle	Évaluer (9)	1

I.3.1.3. Ontologies

Pour illustrer le référencement sémantique, nous avons créé des ontologies contenant les connaissances incluses dans les compétences. Il est à noter que ces ontologies servent uniquement à illustrer le référencement sémantique en montrant

comment une compétence peut être liée à une ontologie par la connaissance sur laquelle elle s'applique. Nos ontologies-exemples sont très petites, mais dans la réalité, des ontologies si restreintes n'existeraient pas. Nous avons créé trois ontologies fictives représentées graphiquement sur les figures qui suivent avec le logiciel MOT+ ONTO dont la légende se trouve à l'appendice A page 173.

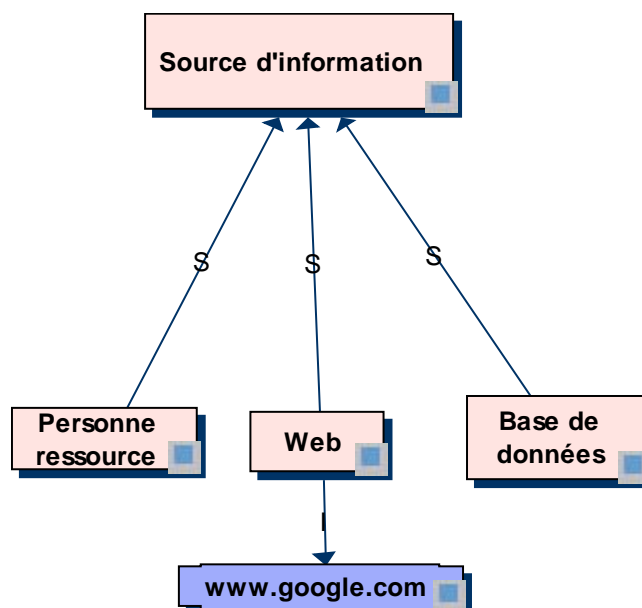


Figure I-16. Ontologie du concept Source d'information
L'ontologie *Source d'information* est liée aux compétences C1 et C4.

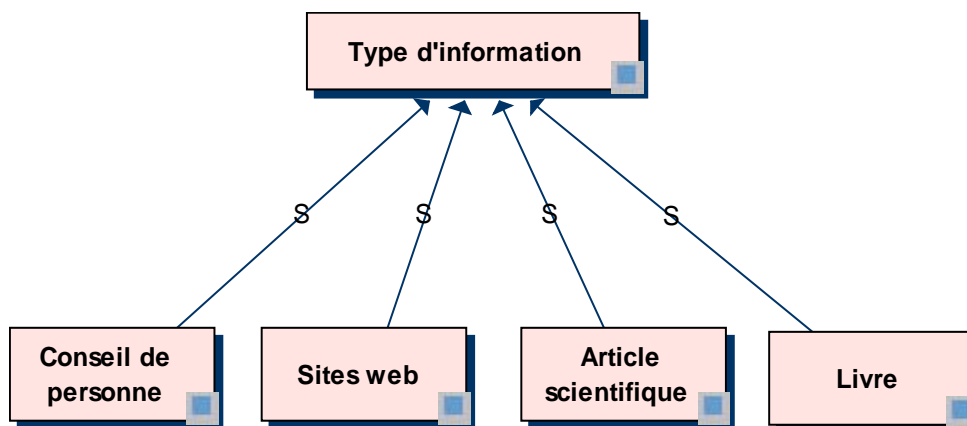


Figure I-17. Ontologie du concept Type d'information
Cette ontologie est liée à la compétence C2.

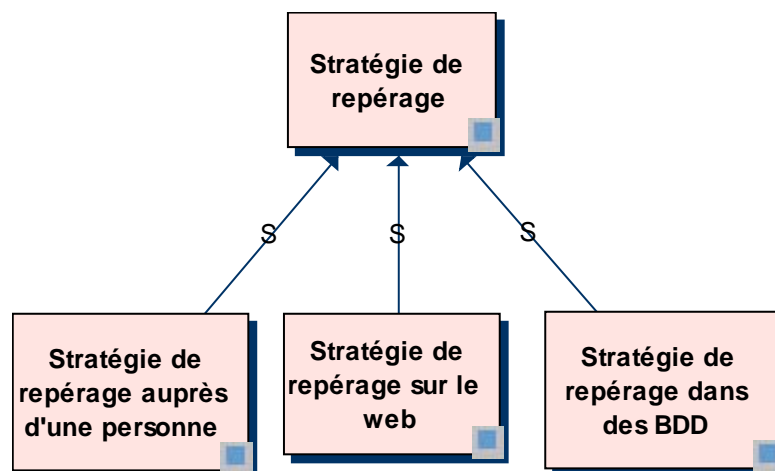


Figure I-18. Ontologie du concept Stratégie de repérage
 Cette ontologie est liée à la compétence C5.

I.3.1.4. Contrat

Voici les valeurs des quatre points constituant un contrat pour ce contexte :

5. **droits des acteurs** : l'acteur dont c'est le modèle et TELOS (le système d'apprentissage en ligne) ont accès au modèle.
6. **granularité de l'évolution** : TELOS peut ajouter des informations ou questionner le modèle en tout temps. À chaque modification du modèle de l'apprenant par TELOS, une nouvelle version est créée.
7. **nombre de points de vue** : le point de vue de l'apprenant et le point de vue consensus qui sont toujours présents.
8. **stratégie de négociation** : c'est la stratégie 3 présentée au chapitre IV page 114 qui est utilisée (c'est-à-dire que les évaluations du système TELOS sont conservées dans le modèle principal).

I.3.2. Scénario

Le scénario de la simulation correspond à un scénario pédagogique créé graphiquement avec TELOS. Ce scénario, comme nous le verrons, comporte deux « chemins ». Cela nous permet de prouver que le lien TELOS- Nosma est bien fonctionnel lors de l'interrogation de Nosma par TELOS. Le contexte utilisé dans le scénario est du type 3 et s'intitule *Contexte cours TELOS*.

I.3.2.1. Acteurs

L'apprenant dont c'est le modèle est Julie Buffet (nom fictif). Ses identifiant et mot de passe (*etu5, password*) lui permettent de se connecter à TELOS qui lui-même se connectera au serveur SVN au cours du scénario pédagogique pour modifier ou interroger le modèle.

I.3.2.2. Production

Une production est nécessaire au bon déroulement de ce scénario. C'est un document texte intitulé *Devoir 1*.

I.3.2.3. Déroulement

Le scénario pédagogique a été créé graphiquement dans TELOS. Il s'intitule *Repérer les informations et les documents* (équivalent du titre du cours). La figure suivante montre le graphe de celui-ci. La flèche verte indique le début du scénario et le carré rouge la fin de celui-ci. Les ovales sont des activités effectuées par l'apprenant ou des fonctions effectuées par TELOS et le losange représente une condition (du style si A alors B). Les valises représentent des données au format texte. La condition comporte deux cas de figure, vrai ou faux. Selon la réponse, le chemin emprunté ne sera pas le même. Expliquons plus en détail les étapes de déroulement du scénario.

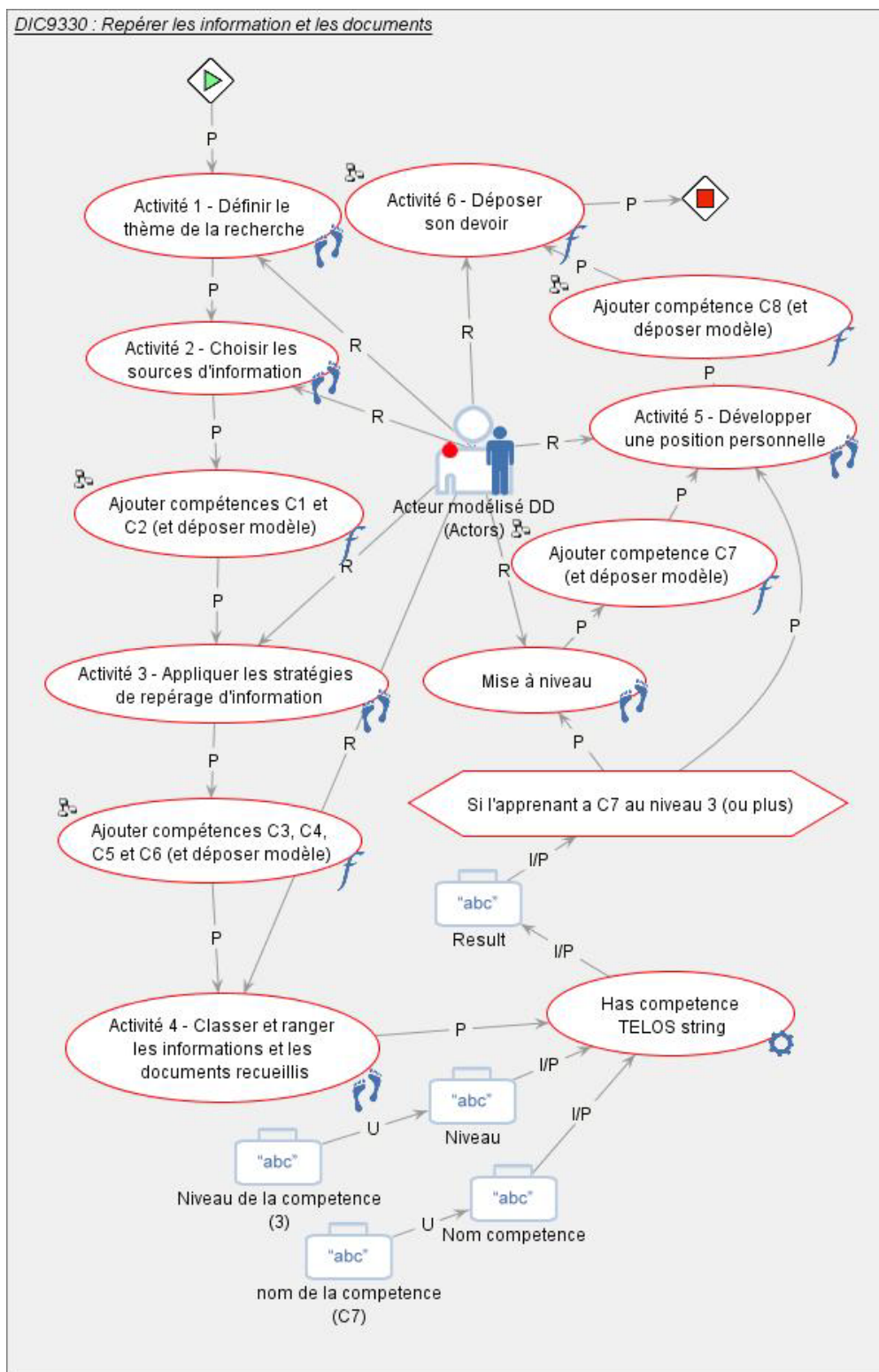


Figure I-19. Scénario pédagogique simulation 3

Le scénario comporte 11 étapes.

1. Création du modèle de l'apprenant.
2. L'apprenant effectue l'activité 1.
3. L'apprenant effectue l'activité 2.
4. TELOS ajoute automatiquement les compétences C1 et C2 et dépose le modèle sur le serveur SVN (une nouvelle version est créée).
5. L'apprenant effectue l'activité 3.
6. TELOS ajoute automatiquement les compétences C3, C4, C5 et C6 et dépose le modèle sur le serveur SVN (une nouvelle version est créée).
7. L'apprenant effectue l'activité 4.
8. TELOS effectue un test pour savoir si la compétence C7 est dans le modèle de l'apprenant au niveau 3 ou supérieur;
 - 8a. Si non :
 l'apprenant effectue la mise à niveau;
 TELOS ajoute automatiquement la compétence C7 et dépose le modèle sur le serveur SVN (une nouvelle version est créée).
 - 8b. Si oui : passer directement à l'étape 9.
9. L'apprenant effectue l'activité 5.
10. TELOS ajoute automatiquement la compétence C8 et dépose le modèle sur le serveur SVN (une nouvelle version est créée).
11. L'apprenant effectue l'activité 6 (ajouter une production dans son modèle).

I.3.3. Simulation

Le modèle de l'apprenant Julie Buffet est modifié au fur et à mesure du déroulement du scénario. Pour tester le bon fonctionnement de la fonction d'interrogation du modèle de l'apprenant (*hasCompetence*), nous avons effectué deux fois le scénario. Dans le cas A, le modèle de l'apprenant Julie Buffet est vide au départ, l'apprenant devra ainsi effectuer la mise à niveau avant de passer à l'activité 5. Dans le cas B, la compétence C7 se trouve dans le modèle de Julie au niveau 3 dès

le début du scénario, ainsi la mise à niveau ne sera pas effectuée et TELOS n'ajoutera pas la compétence C7 dans le modèle de Julie. Cette simulation vise aussi à montrer le bon fonctionnement des fonctions de modifications du modèle, soit l'ajout d'une compétence (étape 4) et l'ajout d'une production (étape 11).

I.3.3.1. Preuve de l'ajout d'une compétence (étape 4)

TELOS ajoute les compétences C1 et C2 à l'étape 4. La figure I-20 montre le modèle de Julie à cette étape.

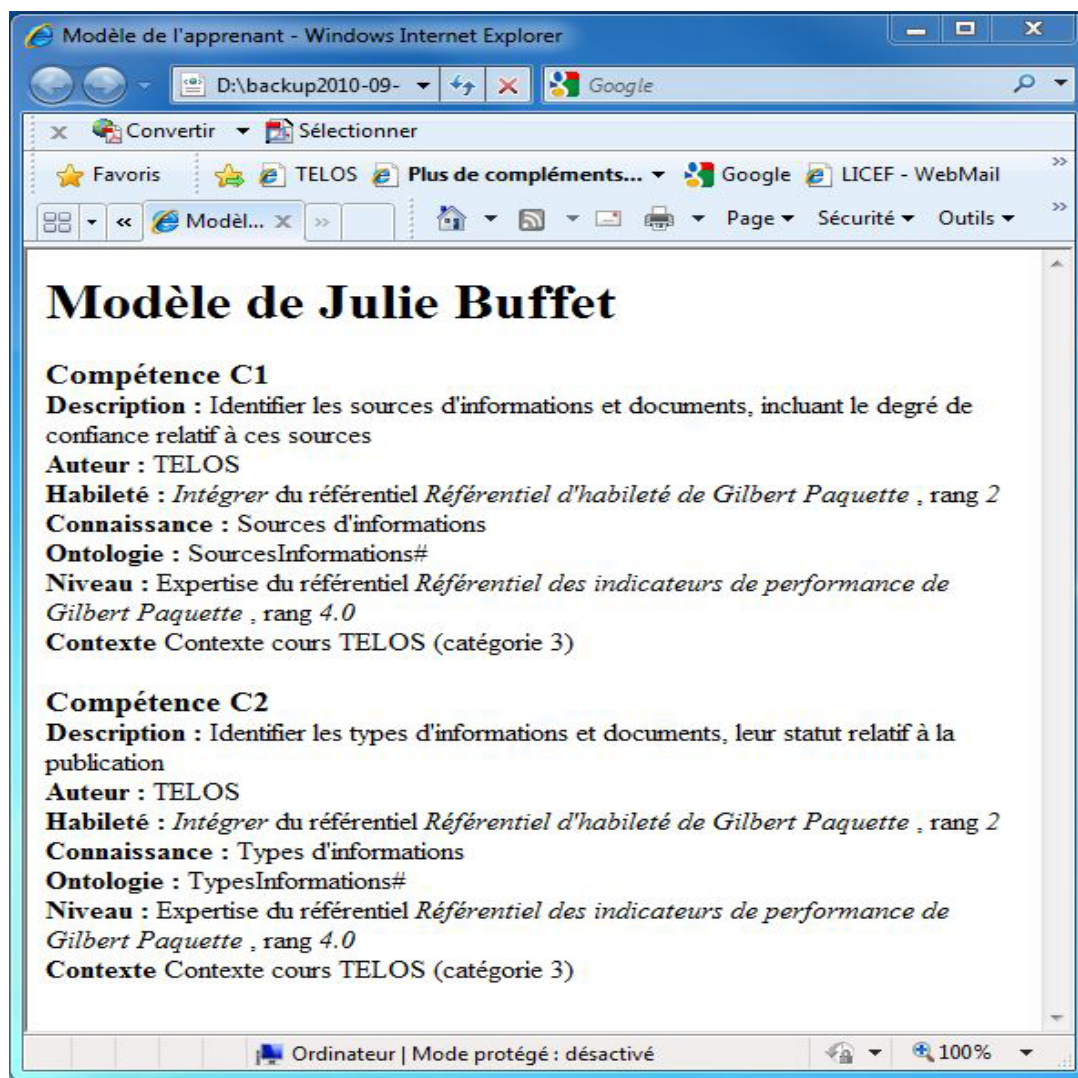


Figure I-20. Modèle de l'apprenant simulation 3 étape 4 (cas A)

La figure I-21 montre le modèle de Julie à la même étape mais dans le cas B, où C7 est déjà présente dans le modèle (ajouté par le professeur Robitaille dans le cadre du cours ABC).



Figure I-21. Modèle de l'apprenant simulation 3 étape 4 (cas B)

I.3.3.2. Preuve de ajout production (étape 11)

À l'étape 11, Julie ajoute une production à son modèle. La figure I-22 montre que la production y est bien présente (ou plus précisément la fiche de métadonnées concernant la production). Remarquons que l'adresse du fichier est une adresse TELOS, en effet, ce système permet de stocker des fichiers et les apprenants peuvent téléverser leurs productions sur le serveur de TELOS.

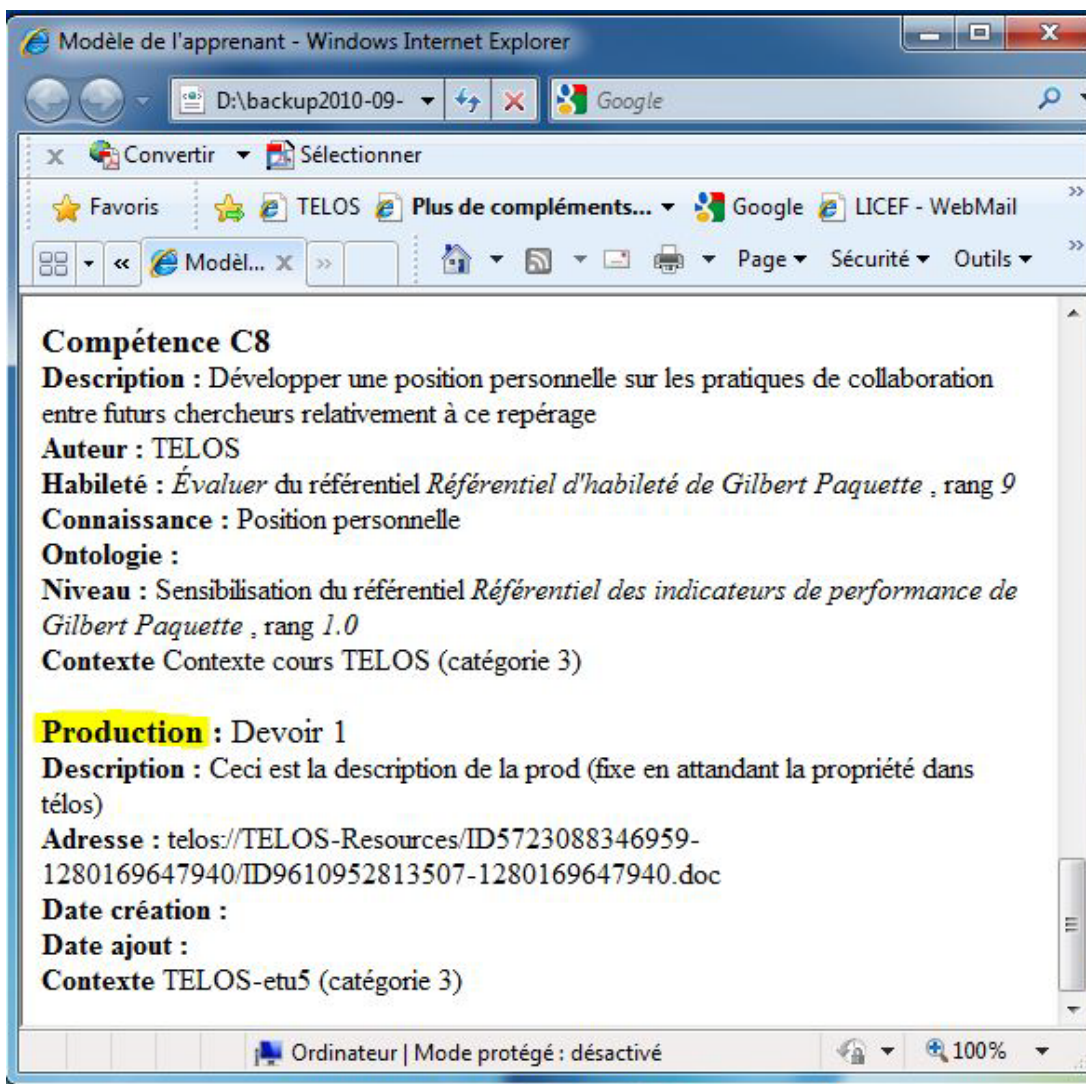


Figure I-22. Modèle de l'apprenant simulation 3 étape 11

Les deux étapes présentées (4 et 11) montrent aussi que les fonctions permettant de récupérer et de déposer le modèle sur le serveur SVN fonctionnent. En

effet, sans elles, il ne serait pas possible de voir les modifications apportées au modèle.

I.3.3.3. Preuve de l'interrogation du modèle (étape 8)

TELOS offre un environnement utilisateur. Pour qu'un apprenant puisse commencer un scénario pédagogique, celui-ci doit d'abord être « lancé » par un utilisateur administrateur. Nous avons fait la simulation avec un seul étudiant mais le scénario est développé génériquement, ce n'est qu'à son instanciation dans TELOS (au moment où il est lancé) que l'apprenant concerné est précisé. La figure I-23 montre la fenêtre de TELOS qui permet à l'administrateur de dire que le scénario pédagogique est lancé pour l'apprenant Julie Buffet. Nous voyons en arrière fond le nom du scénario, *Repérer les informations et les documents* ainsi qu'une case à remplir pour le champ *Acteur modélisé*.

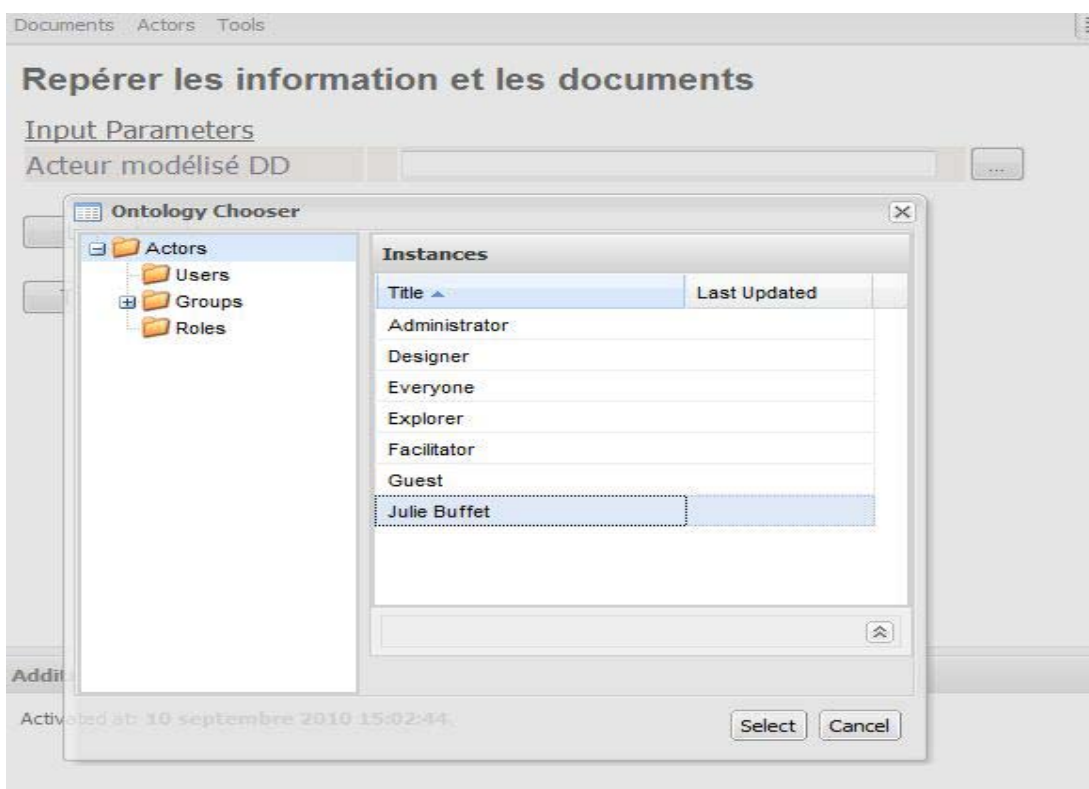


Figure I-23. Instanciation du scénario pédagogique pour Julie Buffet

Voyons maintenant ce qui se passe dans le cas A, celui où la mise à niveau doit être effectuée. Pour cela, la figure I-24 nous montre l'interface TELOS de Julie au cours du scénario.

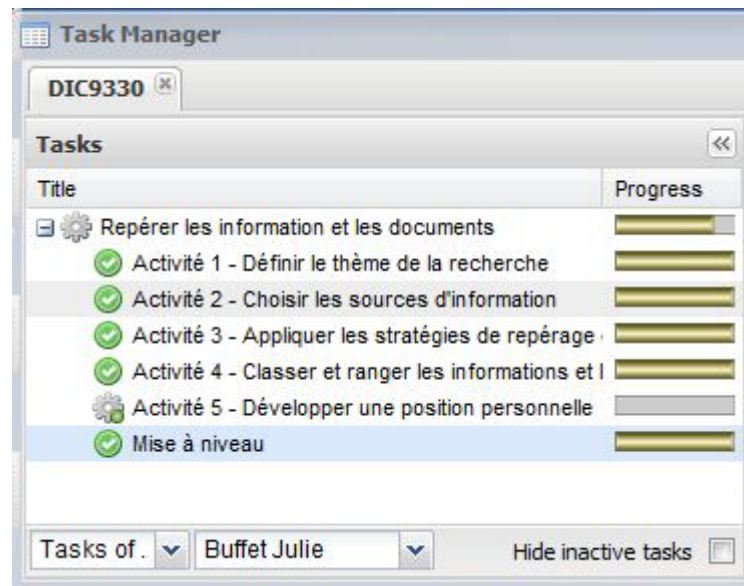


Figure I-24. Interface TELOS pour l'apprenant (simulation 3 cas A)

Nous voyons sur la figure que les activités 1 à 4 ont été effectuées ainsi que la mise à niveau (elles sont validées par une coche verte). L'apprenant en est à l'activité 5 qu'il doit effectuer avant de passer à la suite du scénario. Notons que l'activité 5 apparaît avant la mise à niveau parce que TELOS affiche les activités par ordre alphabétique. Mais nous voyons bien que la mise à niveau a été effectuée avant que l'activité 5 ne le soit.

Dans le cas B, la mise à niveau n'apparaîtra pas sur l'interface de l'apprenant puisque celui-ci n'a pas à l'effectuer (voir figure I-25). La figure I-26 montre l'interface de l'administrateur qui peut suivre le parcours de l'apprenant dans le scénario. Nous voyons que la mise à niveau et que l'ajout de la compétence C7 restent grisées même si l'activité 5 a été faite. Aussi, nous voyons ici que sur l'interface de l'administrateur, toutes les activités du scénario apparaissent, celles effectuées par l'apprenant, celles exécutées par le système TELOS et celles qui ne seront pas exécutées (comme la mise à niveau par exemple).

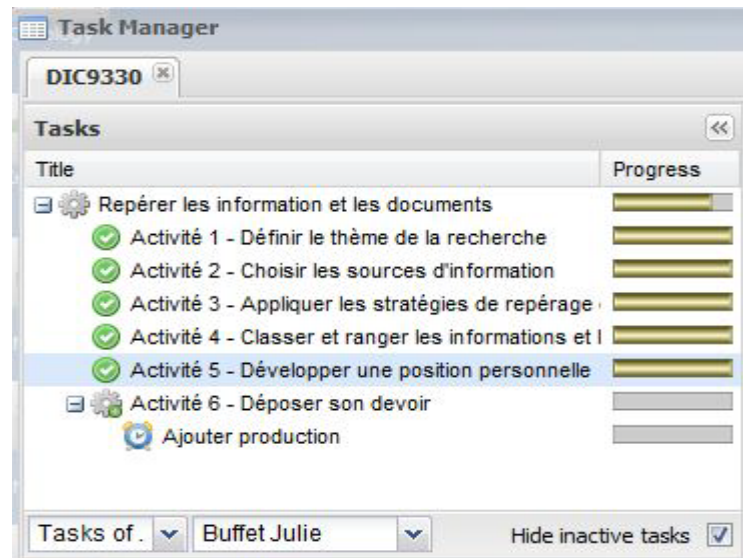


Figure I-25. Interface TELOS pour l'apprenant (simulation 3 cas B)

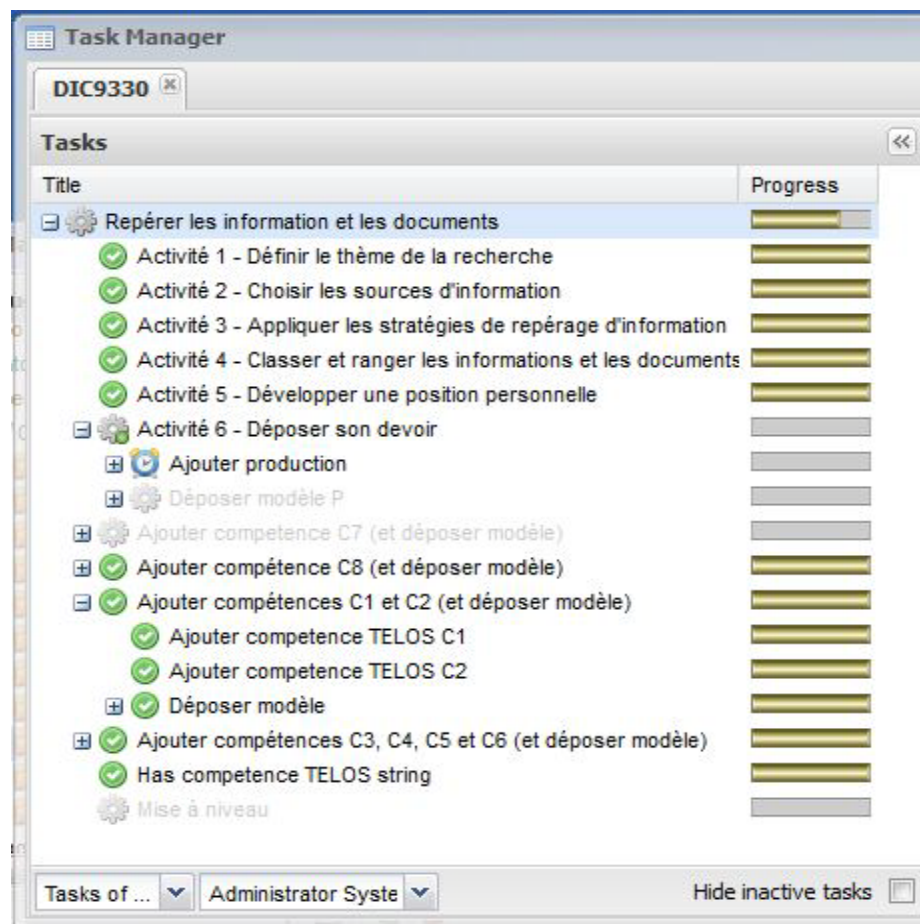


Figure I-26. Interface TELOS pour l'administrateur (simulation 3 cas B)

GLOSSAIRE / LEXIQUE

Modèle de l'apprenant principal : modèle qui est gardé en mémoire avec deux points de vue.

Modèle contexte : modèle pour un cours particulier, avec les référentiels de compétence et d'habiletés.

Modèle de l'apprenant contexte temporaire : modèle contexte d'un cours mais pour un apprenant particulier, lié à un modèle principal.

Point de vue consensus : dans le cadre du multi points de vue, modèle qui a le consensus.

LISTE DE RÉFÉRENCES

- Anderson, P. (2007). What is Web 2.0 ? Ideas, technologies and implications for education. Repéré à <http://www.jisc.ac.uk/publications/reports/2007/twwweb2.aspx>.
- Anderson, T. et Whitelock, D. (2004). The Educational Semantic Web: Visioning and practicing the Future of Education. *Journal of Interactive Media in Education*, 1 (Special Issue on the Educational Semantic Web), 1-15.
- Ankolekar, A., Krötzsch, M., Tran, D. T. et Vrandečić, D. (2008). The Two Cultures : Mashing up Web 2.0 and the Semantic Web. *Journal of Web Semantics*, 6(1), 70–75.
- Aroyo, L. et Dicheva, D. (2004). The new Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society*, 7(4), 59-69.
- Ayala, G. et Paredes, R. (2003). Learner model servers: personalization of web based educational applications based on digital collections. Communication présentée ED-MEDIA 2003, World Conference on Educational Multimedia, Hypermedia and Telecommunications, Honolulu, USA.
- Bachman, F., Bass, L., Buhman, C., Comella-Dorda, S., Long, F., Robert, J. (2000). Volume II: Technical Concepts of Component-Based Software Engineering.
- Baker, M. J. (2000). The roles of models in Artificial Intelligence and Education research: a prospective view. *International Journal of Artificial Intelligence and Education*, 11, 122-143.
- Balacheff, N., Baron, M., Desmoulins, C., Grandbastien, M. et Vivet, M. (1997). Conception d'environnements interactifs d'apprentissage avec ordinateur - Tendances et perspectives. *PRC-GDR IA*, 315-337.
- Barab, S. A. et Duffy, T. (2000). From practice fields to communities of practice. Dans D. Jonassen et S. M. Land (dir.), *Theoretical Foundations of Learning Environments* (p. 25-56). Mahwah, NJ: Lawrence Erlbaum Associates.
- Baron, M. (1994). EIAO, quelques repères. *Terminal, technologie de l'information, culture et société*, 65, 67-84.
- Barrett, H. (2004). Electronic Portfolios as Digital Stories of Deep Learning – Emerging Digital Tools to Support Reflection in Learner-Centered Portfolios. Repéré le 8 July 2008 à <http://electronicportfolios.org/digistory/epstory.html>.

- Benchikha, F., Boufaïda, M. et Seinturier, L. (2005). Viewpoints: a framework for object oriented database modelling and distribution. *Data Science Journal*, 4, 92-107.
- Berlanga, A. et García, F. (2004). A Proposal to Define Adaptive Learning Designs. Communication présentée Workshop on Applications of Semantic Web Technologies for Educational Adaptive Hypermedia (SW-EL 2004) in AH 2004, Eindhoven, Netherlands.
- Berners-Lee, T., Hendler, J. et Lasilla, O. (2001). The Semantic Web. *Scientific American*, 284 (5).
- Better SCM Initiative (2007). Version Control System Comparison.
- Beugnard, A., Jézéquel, J.-M., Plouzeau, N. et Watkins, D. (1999). Making components contract aware. *IEEE Computer*, 13(7).
- Bézivin, J. et Gerbé, O. (2001). Towards a precise definition of the OMG/MDA framework. Communication présentée IEEE conference automated software engineering (ASE'2001), San Diego, USA.
- Bloom, B. S. et al. (1969). *Taxonomie des objectifs pédagogiques: domaine cognitif*. (Traduit par M. Lavallée). Montréal: Education Nouvelle.
- Bortzmeyer, S. (2005). Les nouveaux systèmes de gestion de version: Journées Réseaux 2005. Repéré à <http://2005.jres.org/paper/2.pdf>.
- Brooks, C., Winter, M., Greer, J. et McCalla, G. (2004). The Massive User Modelling System (MUMS). Communication présentée Intelligent Tutoring Systems 2004, Maceió, Alagoas, Brazil.
- Brown, J. S., Collins, A. et Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-42.
- Bruillard, E. (1997). *Les machines à enseigner*. Paris: Hermès.
- Bruillard, E., Delozanne, E., Leroux, P., Delannoy, P., Dubourg, X., Jacoboni, P. (2000). Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM. *Sciences et Techniques Educatives*, 7(1), 87-145.
- Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. Special Issue on Intelligent Systems and Teleteaching, *Künstliche Intelligenz*, 4, 19-25.

- Brusilovsky, P. (2000). Concept-based courseware engineering for large scale Web-based education. Communication présentée WebNet'2000, World Conference of the WWW and Internet, San Antonio, TX.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11, 87-110.
- Brusilovsky, P. (2003). Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology*, 34(4), 487-497.
- Buzzetto-More, N. (Dir.). (2010). *The E-Portfolio Paradigm: Informing, Educating, Assessing, and Managing With E-Portfolios*. Santa Rosa, California: Informing Science Press.
- Cambridge, D. (2010). *E-Portfolios for Lifelong Learning and Assessment*. San Francisco: Jossey-Bass.
- Cambridge, D., Cambridge, B. L. et Yancey, K. B. (2009). *Electronic portfolios 2.0: emergent research on implementation and impact*. Sterling: Stylus.
- Castor. <http://www.castor.org/index.html>.
- Cheal, C. (2010). Implications of open source e-portfolios. Dans N. Buzzetto-More (dir.), *The E-portfolio paradigm: informing, educating, assessing and managing with E-portfolios* (p. 35-62). Santa Rosa, California: Informing Science Press.
- Cloutier, M., Fortier, G. et Slade, S. (2006). *Le portfolio numérique Un atout pour le citoyen apprenant*.
- Collins-Sussman, B., Fitzpatrick, B. et Pilato, C. M. (2002). *Version Control with Subversion*. Sebastopol: O'Reilly Media.
- Conradi, R. et Westfechtel, B. (1998). Version models for software configuration management. *ACM computing surveys*, 30(2), 232-282.
- Cooley, W. W. et Lohnes, P. R. (1976). *Evaluation research in education*. New York: Halsted Press.
- Dagger, D., Wade, V. et Conlan, O. (2003). Towards “anytime, anywhere” Learning: The Role and Realization of Dynamic Terminal Personalization in Adaptive eLearning. Communication présentée World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003, Hawai. Repéré à <http://dl.aace.org/12699>.

- Dagger, D., Wade, V. et Conlan, O. (2004). Developing Active Learning Experiences for Adaptive Personalised eLearning. Communication présentée International Conference Adaptive Hypermedia 2004, Eindhoven.
- De Vries, E. et Baillé, J. (2006). Apprentissage: référents théoriques pour les EIAH. Dans M. GrandBastien et J.-M. Labat (dir.), Environnements informatiques pour l'apprentissage humain (p. 27-46). Paris: Hermès/Lavoisier.
- Denaux, R., Aroyo, L. et Dimitrova, V. (2005). An Approach for Ontology-based Elicitation of User Models to Enable Personalization on the Semantic Web. Communication présentée International World Wide Web Conference, Chiba, Japan.
- Devedzic, V. (2006). Semantic web and education. Berlin Heidelberg New York: Springer.
- Devedzic, V., Jovanovic, J. et Gasevic, D. (2007). The Pragmatics of Current E-Learning Standards. IEEE Internet Computing, 11(3), 19-27.
- DiBiase, D. (2002). Using e-Portfolios at Penn State to Enhance Student Learning - Status, Prospects, and Strategies.
- Dieng-Kuntz, R., Corby, O., Grandon, F., Giboin, A., Golebiowska, J., Matta, N. et Ribière, M. (2000). Méthodes et outils pour la gestion des connaissances. Une approche pluridisciplinaire du Knowledge Management. Paris: Dunod.
- Dolog, P., Henze, N., Nejdl, W. et Sintek, M. (2004, 17-22 May). Personalization in distributed e-learning environments. Communication présentée International World Wide Web Conference, New York, USA.
- ePortConsortium. (2003). Electronic Portfolio White Paper: ePortConsortium.
- Eyssautier-Bavay, C. (2004). Le portfolio en éducation, concept et usages. Communication présentée Colloque TICE Méditerranée, Nice, France.
- Favre, J. M., Estublier, J. et Blay-Fornarino, M. (2006). L'ingénierie dirigée par les modèles, au-delà du MDA. Paris: Editions Lavoisier. Hermes Sciences Publications.
- Finkelstein, A. et Sommerville, I. (1996). The Viewpoints FAQ. Software Engineering Journal, 11, 2-4.
- Garlatti, S. et Prié, Y. (2004). Adaptation et personnalisation dans le web sémantique. Revue i3.org, 24.

- Global Learning Consortium, I. (2001). IMS Learner Information Packaging Best Practice & Implementation Guide, Final Specification Version 1.0: IMS Global Learning Consortium.
- Global Learning Consortium, I. (2002). IMS Reusable Definition of Competency or Educational Objective – Information Model, Version 1.0 Final Specification: IMS Global Learning Consortium.
- Global Learning Consortium, I. (2003). IMS Learning Design XML Binding, V1, Specification. : IMS Global Learning Consortium.
- Global Learning Consortium, I. (2005). IMS ePortfolio Best Practice and implementation Guide, Version 1.0 Final Specification: IMS Global Learning Consortium.
- Global Learning Consortium, I. (2006). IMS Question and Test Interoperability Overview, V2.1, Specification.: IMS Global Learning Consortium.
- Hansen, C. et McCalla, G. (2003). Active Open Learner Modelling. Communication présentée AIED2003, Sydney.
- Hibou, M. et Py, D. (2006). Représentation des connaissances de l'apprenant. Dans M. Grandbastien et J. M. Labat (dir.), Environnements informatiques pour l'apprentissage humain (p. 97-116). Paris: Lavoisier.
- Hotte, R. (2009). Ingénierie d'un EIAH en ligne orienté compétences. Communication présentée EIAH 2009, Université du Maine, Le Mans (France).
- Hotte, R. et Contamines, J. (2007). Communautés de pratique : auteures et utilisatrices des banques de ressources éducatives. Dans M. Baron, D. Guin et L. Trouche (dir.), Environnements informatisés et ressources numériques pour l'apprentissage. Conception et usages, regards croisés (p. 669-682). Paris: Hermès-Lavoisier.
- HR-XMLConsortium. (2007). Resume Recommendation: Human Resources XML Consortium.
- IEEE, L. T. S. C. (2002). Draft Standard for Learning Object Metadata (LOM). New York: IEEE.
- International Organization Standardization. (1998). ISO 9241-11 Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) – Partie 11: lignes directrices relatives à l'utilisabilité.
- Internet Mail Consortium. (1998). vCard, The Electronic Business Card. Internet Mail Consortium. Repéré le 25 mai 2007 à <http://www.imc.org/pdi>.

- Iqbal, A., Oppermann, R., Patel, A. et Kinshuk, A. (1999). Classification of Evaluation Methods for Intelligent Tutoring Systems. Dans U. Arend, E. Eberleh et K. Pitschke (dir.), *Software Ergonomie '99 Design von Informationswelten* Pitschke (p. 169-181). Leipzig: B. G. Teubner Stuttgart.
- ITU-T. (2001). Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services: International Telecommunication Union – Telecommunication Standardization Sector.
- Jonnaert, P. (2002). *Compétences et socioconstructivisme. Un cadre théorique.* Bruxelles, Belgique: De Boeck.
- Kilpi, T. (1997). New Challenges for Version Control and Configuration Management: a Framework and Evaluation. *IEEE Computer*, 33-41.
- Krathwohl, D., Bloom, B. et Masia, B. (1969). *Taxonomie des objectifs pédagogiques, Tome 2 Domaine affectif.* Montréal: Éducation Nouvelle.
- Larman, C. (2005). *UML 2 et les design patterns: Analyse et conception orientées objet et développement itératif.* Paris: Pearson Education France.
- Lasnier, F. (2000). *Réussir la formation par compétences.* Montréal: Guérin.
- Lave, J. et Wenger, E. (1990). *Situated Learning: Legitimate Peripheral Participation.* Cambridge: Cambridge University Press.
- Le Moigne, J. L. (1990). *La Modélisation des systèmes complexes.* Paris: Dunod.
- LORNET. (2005). LORNET project. Repéré le 25 mai 2007 à <http://www.lornet.org/Default.aspx?tabid=384>.
- Magnan, F. et Paquette, G. (2007). TELOS: An Operating System for eLearning Applications. Dans D. C. Pahl (dir.), *Architecture Solutions for E-Learning Systems.* Ireland: Dublin City University, Idea Group Publishing.
- Mariño, O., Rechenmann, F. et Uvietta, P. (1990). Multiple Perspectives and Classification mechanism in object-oriented representation. Communication présentée 9th European Conference on Artificial Intelligence ECAI, Estocolmo.
- Mark, M. A. et Greer, J. E. (1993). Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence in Education (Special Issue on Evaluation)*, 4(2/3), 129-153.
- Mayhew, D. (1999). *The usability engineering lifecycle: a practitioner's handbook for user interface design.* San Francisco: Morgan Kaufmann Publishers.

- McCalla, G., Vassileva, J., Greer, J. et Bull, S. (2000). Active Learner Modelling. Communication présentée ITS 2000: Intelligent Tutoring Systems, Montréal, QC, Canada.
- Meister, L. (2005). Subversion, un système de gestion de version: Note technique CODICIEL 2005-02.
- Mellor, S. J., Scott, K., Uhl, A. et Weise, D. (2004). MDA Distilled: Principles of Model Driven Architecture. Boston: Addison-Wesley.
- Miara, A. (2010). TELOS External Software Component's User Guide: LICEF.
- Miles, M. M. et Huberman, A. M. (2003). Analyse des données qualitatives, traduction de la 2e édition américaine. Bruxelles: De Boeck.
- Millard, D. E. et Ross, M. (2006). Web 2.0: hypertext by any other name? Communication présentée Seventeenth conference on Hypertext and hypermedia, Odense, Denmark.
- Ministère de l'Éducation du Loisir et du Sport du Québec (2006). Échelles des niveaux de compétence.
- Moulet, L. (2006b). Rapport d'analyse d'Open Source Portfolio. Montréal: Licef, Téluc-UQAM.
- Moulet, L. (2006a). Revue de littérature de l'ePortfolio : Définitions, contenus et usages. Visant à l'intégration d'un ePortfolio dans le modèle de l'apprenant d'un système d'apprentissage en ligne. . Montréal: LICEF, Télé-université.
- Murray, T. (1993). Formative Qualitative Evaluation for "Exploratory" ITS research. Journal of Artificial Intelligence in Education (Special Issue on Evaluation), 4(2/3), 179-207.
- Nicaud, J. F. et Vivet, M. (1988). Les Tuteurs Intelligents : réalisations et tendances de recherche. Technique et Science Informatiques, 7(1), 21-45.
- Nielsen, J. (1993). Usability engineering. Boston: Academic Press Professional.
- Nkambou, R., Bourdeau, J. et Mizoguchi, R. (Dir.). (2010). Advances in Intelligent Tutoring Systems. Pittsburg, USA: Springer.
- Nodenot, T. (2005). Contribution à l'ingénierie dirigée par les modèles en EIAH : le cas des situations-problèmes coopératives: Thèse d'habilitation. Université de Pau et des Pays de l'Adour.

- Nogier, J.-F. (2008). *Ergonomie du logiciel et design web : Le manuel des interfaces utilisateur*, 4ème édition. Paris: Dunod.
- Object Management Group. Model Driven Architecture. Repéré à <http://www.omg.org/mda>.
- Object Management Group. (2002). *Meta-Object Facility (MOF). Specification version 1.4*.
- Office Québécois de la Langue Française. *Grand dictionnaire terminologique*. Repéré à http://www.granddictionnaire.com/btml/fra/r_motclef/index1024_1.asp.
- Open Source Portfolio. <http://www.theospi.org>.
- Paillé, P. et Mucchielli, A. (2003). *L'analyse qualitative en sciences humaines et sociales*. Paris: Armand Colin.
- Papert, S. (Dir.). (1999). *Logo Philosophy and Implementation*. Montreal: LCSi.
- Paquette, G. (2002b). *L'ingénierie pédagogique*. Sainte-Foy, Québec: Presses de l'Université du Québec.
- Paquette, G. (2002a). *Modélisation des connaissances et des compétences: Un langage graphique pour concevoir et apprendre*. Sainte-Foy, Québec: Presse de l'Université du Québec.
- Paquette, G. (2007). *An Ontology and a Software Framework for Competency Modeling and Management*. *Educational Technology and Society, Special issue in Advanced Technologies for Life-Long Learning*, 10(3), 1-21.
- Paquette, G., Bourdeau, J., Henri, F., Basque, J., Léonard, M. et Maina, M. (2003). *Construction d'une base de connaissances et de ressources sur le téléapprentissage*. *Revue STICEF*, 10, 29-56.
- Paquette, G. et Mariño, O. (2006). *Learning Objects, Collaborative Learning Designs and Knowledge Representation*. *Technology, Instruction, Cognition and Learning*, 3, 1-10.
- Paquette, G. et Rosca, I. (2004). *An Ontology-based Referencing of Actors, Operations and Resources in eLearning Systems*. Communication présentée SW-EL/2004 Workshop, Eindhoven.
- Pernin, J.-P. (2006). *Normes et standards pour la conception, la production et l'exploitation des EIAH*. Dans M. Grandbastien et J.-M. Labat (dir.), *Environnements Informatiques pour l'Apprentissage Humain* (p. 201-222). Paris: Hermès.

- Pernin, J.-P. et Lejeune, A. (2004). Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios. Communication présentée Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie, Compiègne.
- Polsani, R. P. (2003). Use and Abuse of Reusable Learning Objects. *Journal of Digital Information*, 3(4).
- Prax, J.-Y. (2003). *Le Manuel du Knowledge Management: Une approche de 2e génération*. Paris: Dunod.
- Rabardel, P. (1995). *Les Hommes et les Technologies, approche cognitive des instruments contemporains*. Paris: Armand Colin Éditeur.
- Raynal, F. et Rieunier, A. (1997). *Pédagogie : dictionnaire des concepts-clés*. Paris: ESF.
- Razmerita, L., Angehrn, A. et Maedche, A. (2003). Ontology based user modeling for Knowledge Management Systems. Communication présentée 9th International conference on User Modeling, Pittsburgh, USA.
- Research Councils Uk (2005). Skills training requirements for research students: joint statement of the UK research councils.
- Rivière, M. (1999). Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels (thèse). (Université de Nice - Sophia Antipolis).
- Roegiers, X. (2000). *Une pédagogie de l'intégration: compétences et intégration des acquis dans l'enseignement*. Bruxelles: De Boeck Université.
- Scallon, G. (2004). L'évaluation des apprentissages dans une approche par compétences. Montréal: Éditions du Renouveau Pédagogique Inc.
- Scapin, D. L. et Bastien, J. M. C. (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behavior & Information Technology*, 17(4/5), 220-231.
- Seidwitz, E. (2003). What models mean. *IEEE Software*, 20(5), 26-32.
- Self, J. (1990). Theoretical foundations for Intelligent Tutoring Systems. *Journal of Artificial Intelligence in Education*, 1(4), 3-14.
- Self, J. A. (1988). Student models: what use are they? Communication présentée Artificial Intelligence Tools in Education, Amsterdam, North-Holland.

- Selic, B. (2003). The Pragmatics of Model-Driven Development. *IEEE Software*(5), 19-25.
- Shaikh, M. et Cornford, T. (2003). Version Management Tools: CVS to BK in the Linux Kernel. Communication présentée 25th International Conference on Software Engineering - Taking Stock of the Bazaar: The 3rd Workshop on Open Source Software Engineering, Portland, Oregon.
- Sison, R. et Shimura, M. (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education*, 9, 128-158.
- Stewart, A., Niederee, C. et Metha, B. (2004). State of the art in user modeling for personalization in content, service and interaction. NSF/DELOS Report on Personalization.
- Stojanovic, L., Staab, S. et Studer, R. (2001, October 23-27). Elearning based on the Semantic Web. Communication présentée World Conference on the WWW and the Internet (WebNet 2001), Orlando, Florida, USA.
- Stutt, A. et Motta, E. (2004). Semantic Learning Webs. *Journal of Interactive Media in Education*, Special Issue on the Educational Semantic Web., 10, 1-32.
- Subversion. <http://subversion.tigris.org>.
- SVNKit. <http://svnkit.com>.
- Tchounikine, P. (2002). Quelques éléments sur la conception et l'ingénierie des EIAH. Communication présentée Assises du GDR-I3.
- Tchounikine, P. (2009). Précis de recherche en ingénierie des EIAH. Repéré à <http://membres-liglab.imag.fr/tchounikine/Precis.html>.
- Terrasse, M. N., Savonnet, M., Leclercq, E., Grison, T. et Becker, G. (2005). Points de vue croisés sur les notions de modèles et de métamodèles. Communication présentée IDM'05, Premières journées sur l'ingénierie dirigée par les modèles, Paris.
- Torre, I. (2000). A Modular Approach for User Modelling. Communication présentée International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Trento, Italy.
- Treuer, P. et Jenson, J. D. (2003). Electronic Portfolios need standards to thrive. The proliferation of e-portfolio applications requires compatible software and design standards to support lifelong learning. *Educause Quarterly*, 2.

- Tricot, A., Plégat-Soutjis, F., Camps, J.-F., Amiel, A., Lutz, G. et Morcillo, A. (2003). Utilité, utilisabilité, acceptabilité : interpréter les relations entre trois dimensions de l'évaluation des EIAH. Dans C. Desmoulins, P. Marquet et D. Bouhineau (dir.), *Environnements informatiques pour l'apprentissage humain* (p. 391-402). Paris: ATIEF / INRP.
- Van der Maren, J., M. (2003). *La recherche appliquée en pédagogie*. (2^{ee} éd.). Bruxelles: De Boeck.
- Van der Maren, J.-M. (1996). *Méthodes de recherche pour l'éducation*. (2^{ee} éd.). Montréal: Les Presses de l'Université de Montréal.
- Vantroys, T. et Peter, Y. (2003). COW, a Flexible Platform for the Enactment of Learning Scenarios. *Lecture Notes in Computer Science*, 2806, 168-182.
- Vantroys, T. et Peter, Y. (2005). COW, une plate-forme de support d'exécution de scénarios pédagogiques. *Revue STICEF*, numéro spécial Conceptions et usages des plates-formes de formation, 12, 117-156.
- Villalobos, J. (2003). *Fédération de Composants: une Architecture Logicielle pour la Composition par Coordination* (thèse). (Université Joseph Fourier de Grenoble).
- Vygotski, L.-S. (1934-1997). *Pensée et langage*. (3^{ème} éd.). Paris: La Dispute.
- Wang, Y. (2007). The Theoretical Framework of Cognitive Informatics. *Int'l J. of Cognitive Informatics and Natural Intelligence*, 1(1), 1-27.
- Wang, Y. (Dir.). (2010). *Discoveries and Breakthroughs in Cognitive Informatics and Natural Intelligence* (Vol. 2). NY, USA: Information Science Reference.
- Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.
- Wilson, S., Liber, O., Johnson, M., Beauvoir, P., Sharples, P. et Milligan, C. (2006). Personal learning environments: challenging the dominant design of educational systems. Communication présentée 2nd international workshop on learner-oriented knowledge management and KM-oriented learning, in conjunction with ECTEL 06, Crète, Greece.
- Zubizarreta, J. (2009). *The learning portfolio: reflective practice for improving student learning*. San Francisco: Jossey-Bass.

